

VŠB – Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení

Modelování, monitorování a řízení průmyslové výroby v supervisních systémech

Modeling, Monitoring and Control of
Industrial Production in Supervisory Systems

Student:

Bc. Jakub Přikryl

Osobní číslo:

PRI0160

Vedoucí diplomové práce:

doc. Ing. Lenka Landryová, CSc.

Ostrava 2020

Zadání diplomové práce

Student: **Bc. Jakub Přikryl**

Studijní program: N2301 Strojní inženýrství

Studijní obor: 3902T004 Automatické řízení a inženýrská informatika

Téma: **Modelování, monitorování a řízení průmyslové výroby v supervisních systémech**
Modeling, Monitoring and Control of Industrial Production in Supervisory Systems

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Charakterizujte současný stav problematiky modelování, monitorování a řízení výroby v supervisních systémech.
2. Navrhněte úlohu pro monitorování vybrané části procesu modelované v supervisním systému.
3. Úlohu realizujte ve vývojovém prostředí distribuovaného systému řízení.
4. Navrhněte způsob ověření funkce modelu a zdokumentujte jej pro účel výuky předmětu Návrhu procesních systémů.
5. Zhodnoťte dosažené výsledky a navrhněte směry dalšího řešení.

Seznam doporučené odborné literatury:

PAPCUN, Peter, Erik KAJATI a Jiri KOZIOREK, 2018. Human Machine Interface in Concept of Industry 4.0. 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA). IEEE, 2018, 289-296. DOI: 10.1109/DISA.2018.8490603. ISBN 978-1-5386-5102-5. Dostupné také z: <https://ieeexplore.ieee.org/document/8490603/>

PTAK, Richard, J. P. MORGENTHAL a Simon FORGE, 1999. Manager's guide to distributed environments: from legacy to living systems. New York: John Wiley, 1999. ISBN 0-471-19712-2.

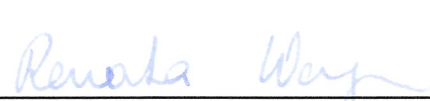
SCHOLTEN, Bianca, 2007. The road to integration: a guide to applying the ISA-95 standard in manufacturing. Research Triangle Park NC: ISA. ISBN 978-0-9792343-8-5.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Lenka Landryová, CSc.**

Datum zadání: 20.12.2019

Datum odevzdání: 18.05.2020


doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry

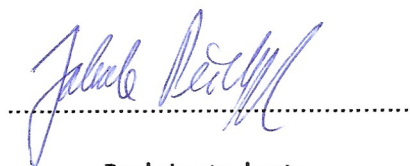



prof. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne 18. května 2020

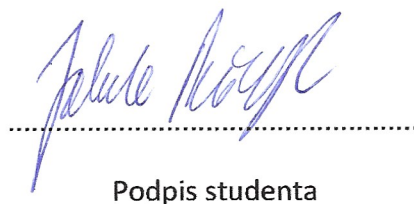


Podpis studenta

Prohlašuji, že:

- jsem si vědom, že na tuto moji závěrečnou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (dále jen Autorský zákon), zejména § 35 (Užití díla v rámci občanských či náboženských obřadů nebo v rámci úředních akcí pořádaných orgány veřejné správy, v rámci školních představení a užití díla školního) a § 60 (Školní dílo),
- беру на вѣдомі, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo užít tuto závěrečnou diplomovou práci nekomerčně ke své vnitřní potřebě (§ 35 odst. 3 Autorského zákona),
- bude-li požadováno, jeden výtisk této diplomové) práce bude uložen u vedoucího práce,
- s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 Autorského zákona,
- užít toto své dílo, nebo poskytnout licenci k jejímu využití, mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše),
- беру на вѣдомі, že podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů - že tato diplomová) práce bude před obhajobou zveřejněna na pracovišti vedoucího práce a v elektronické podobě uložena a po obhajobě zveřejněna v Ústřední knihovně VŠB-TUO, a to bez ohledu na výsledek její obhajoby.

V Ostravě dne 18. května 2020



Podpis studenta

ANOTACE DIPLOMOVÉ PRÁCE

PŘIKRYL, J. *Modelování, monitorování a řízení průmyslové výroby v supervizích systémech: diplomová práce*. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta Strojní, Katedra automatizační techniky a řízení, 2020, 113 s. Vedoucí práce: Landryová, L.

Diplomová práce je věnována tématice řízení průmyslové výroby pomocí supervizních systémů. Čtenář bude seznámen se základními pojmy souvisejícími s modelováním, řízením a vizualizací průmyslových procesů a získá přehled o souvislostech v dynamicky se rozvíjející oblasti supervizního řízení. Diplomová práce shrnuje postupy určené k základní konfiguraci vybraného distribuovaného systému řízení, popisuje architekturu distribuovaného systému ve virtuálním prostředí a uvádí nutné kroky pro přípravu hardwarových komponent systému. Dále jsou představeny softwarové komponenty vybraného distribuovaného systému sloužící k tvorbě kontrolní logiky a operátorských obrazovek. Nabyté poznatky jsou demonstrovány na vzorové demo úloze sloužící ke znázornění současných možností řízení pomocí distribuovaného systému řízení.

ANNOTATION OF MASTER THESIS

PŘIKRYL, J. *Modeling, Monitoring and Control of Industrial Production in Supervisory Systems: Master Thesis*. Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2020, 113 p. Thesis head: Landryová, L.

This master thesis is focused on the topic of industrial production control assisted by supervisory systems. The reader will get to know the basic concepts connected to modelling and visualization of production processes and will get an overview of the dynamically expanding area of supervisory control. This master thesis summarizes the methods for basic configuration of a selected distribution control system, describes the system's virtualized architecture, and shows the fundamental steps for hardware preparation. In the next part of the thesis, software components of system for control logic and operators workplace development are introduced. Acquired skills and knowledge are therefore applied to the creation of demo application focused on representation of distributed control systems possibilities.

Obsah

Seznam použitých zkratk	8
Úvod	9
1 Současný stav techniky supervizního řízení	12
1.1 Distribuovaný systém řízení	12
1.2 Vývoj HMI	13
1.3 Základní funkce SCADA systémů	16
1.4 Přehled dostupných SCADA a DCS řešení	18
1.5 Virtualizace	21
2 Popis původního modelového prostředí demo úloh	23
2.1 Popis úloh	23
2.2 Hardwarové konfigurace	24
2.3 Využívaný software	33
3 Popis nového modelového prostředí	35
3.1 Distribuovaný systém řízení ABB 800xA	35
3.2 Síťová architektura, virtuální počítače a systém 800xA	36
3.3 Control Panel 800	38
4 Nová demo úloha	39
5 Realizace demo úlohy – Příprava a nastavení Hardwaru	45
6 Realizace demo úlohy – Proměnné a vývojový diagram	49
6.1 PLC 1 – Aplikace Crane Motor	50
6.2 PLC 2 – Aplikace Rozvaděč	56
7 Realizace demo úlohy – Programování v Control Builderu	57
7.1 Proměnné	57
7.2 Programovací bloky Function Diagram a Control Module	59
7.3 Program	60
7.4 Řízení úloh – Task Properties	61
7.5 Testovací mód, online mód a nahrání programu do PLC	62
8 Realizace demo úlohy – Testování a ověření funkce	63
9 Základní konfigurace systému 800xA	68
9.1 Vytvoření nového operátorského pracoviště	69
9.2 Konfigurace operátorského pracoviště	71

9.4	Nastavení aplikačního panelu a status panelu	72
9.5	Bezpečnostní nastavení	73
10	Funkční struktura a program Graphics Builder	75
10.1	Základní přehled programu Graphics Builder	76
10.2	Aktivní prvky	77
10.3	Pasivní prvky	78
10.4	Výrazy a proměnné	79
10.5	Odkaz na aspekty	80
11	Vizualizační obrazovky Demo úlohy pro operátorskou stanici	81
11.1	Hlavní menu	81
11.2	Kontrola jeřábu – hlavní panel	82
11.3	Nastavení os a motorů	85
11.4	Obrazovka s nápovědou	87
12	Dotykový panel a Program Panel Builder	88
12.1	Vytvoření nového projektu a vývojového prostředí	89
12.2	Propojení proměnných z PLC a v programu Panel Builder tagů	91
12.3	Pasivní a aktivní prvky vizualizační obrazovky	93
12.4	Nahrání aplikace do panelu	95
13	Vizualizační obrazovky pro dotykový panel PP874	96
13.1	Obrazovka Remote	97
13.2	Obrazovka Manual	98
13.3	Obrazovky Points, Map a Help	99
14	Bezpečnost a uživatelská příručka	101
14.1	Bezpečnostní pravidla demo úlohy	101
14.2	Uživatelská příručka	102
14.3	Manuální režim	103
15	Závěr	104
16	Seznam použité literatury	108
17	Seznam Příloh	113

Seznam použitých zkratek

AI	Artificial Intelligence	Umělá inteligence
APC	Advanced Process Control	Pokročilé řízení procesů
DCS	Distributed Control Systems	Distribuovaný systém řízení
DTC	Direct Torque Control	Přímé řízení momentu
EOW	Extended Operator Workplace	Rozšířené operátorské pracoviště
FBD	Function Block Diagram	Diagram funkční bloků
FM	-	Frekvenční měnič
HMI	Human Machine Interface	Rozhraní člověk - stroj
I/O	Input/Output	Vstup/Výstup
IL	Instruction List	Seznam instrukcí
IPC	Industrial PC	Průmyslový počítač
KPI	Key Performance Indicators	Klíčové ukazatele výkonnosti
LD	Ladder Diagram	Žebříkový diagram
OLE	Object Linking and Embedding	Spojování a vkládání objektů
OPC	OLE for Process Control	OLE pro řízení procesů
PC	Personal Computer	Osobní počítač
PLC	Programmable Logic Controller	Programovatelný logický automat
SCADA	Supervisory Control And Data Acquisition	Supervizní řízení a sběr dat
SFC	Sequential Function Chart	Sekvenční funkční graf
SOA	Service Oriented Architecture	Architektura s orientací na služby
ST	Structured Text	Strukturvaný text
VM	Virtual Machine	Virtuální stroj
VR	Virtual Reality	Virtuální realita

Úvod

Prudký rozvoj výpočetní techniky ovlivnil v průběhu druhé poloviny minulého století většinu oblastí lidské činnosti a informační technologie si našli cestu do drtivé většiny odvětví průmyslové výroby. Zvyšující se výpočetní výkon, miniaturizace a možnosti síťového propojení jednotlivých zařízení otevřeli cestu k implementaci programovatelných automatů (PLC), či průmyslových počítačů (IPC) do výrobních podniků. V počátcích nahrazovali počítače reléová zapojení v centralizovaných velínech a s příchodem standardizovaných komunikačních technologií pak byl jen malý krok k distribuované struktuře řízení. Rozvoj průmyslových síťových prvků a komunikačních protokolů otevřel cestu k robustní hierarchické struktuře řízení. [1]

Současné podniky mnohdy využívají různé softwarové řešení od rozličných výrobců k pokrytí potřeb jednotlivých oddělení. Přestože existuje několik mamutích systémů, které si kladou za cíl postihnout celou problematiku řízení podniků, není výjimkou, že informační, skladový či výrobní systém dodávají koncovým uživatelům různí výrobci. Z důvodu potřeby regulace a vzájemného porozumění mezi výrobcí softwaru, hardwaru a koncovými uživateli řídicích a informačních systémů vznikla série norem popisující úrovně řízení a vztahy mezi nimi. Jedna z nejpodstatnějších norem ISA-95 si klade za cíl poskytnout konzistentní terminologii, které stanoví základy pro komunikaci mezi dodavateli a uživateli technologií, definuje vzorové modely předávání informací a hraje roli průvodce na cestě za aplikační funkcionalitou. Norma dále dělí pomyslnou automatizační pyramidu na pět úrovní, kdy na vrcholné čtvrté úrovni stojí plánovací a strategický informační systém, níže na třetí úrovni pak výrobní systémy a pod nimi přímá kontrola výroby. Na úrovni první najdeme vstupně/výstupní moduly a na nejnižší nulté úrovni se nachází prostředky instrumentace v podobě senzoriky a aktuátorů. Supervizní řízení výroby se nachází na třetí úrovni a tudíž na pomezí mezi technologií a celopodnikovými informačními systémy. [2]

Samotný princip řízení procesů formou supervize stojí na pomezí manuálního a plně automatického řízení. U manuálního řízení je počítač využíván pouze jako sdělovací a ovládací prostředek pro operátora, který má plnou kontrolu nad všemi částmi procesu. Na opačné straně pak stojí autonomní řízení, kdy je operátor v roli pouhého pozorovatele a v zásadě nemá možnost řízení procesu ovlivnit. V případě supervizního řízení dostává operátor všechny informace potřebné k rozhodování v přehledné formě a na jejich základě rozhoduje o dalším průběhu procesu. Velké množství dat je zpracováno na pozadí a operátorovi jsou prezentována pouze vybraná data podstatná pro řízení procesu. [1]

Supervizní řízení je často realizováno pomocí SCADA (Supervisory Control And Data Acquisition) systémů, které v současnosti poskytují možnosti propojení hardwarových zařízení od různých výrobců a jejich kontrolu z grafického prostředí operátora. SCADA systémy využívají databázové servery k ukládání dat a jejich následnému vyhodnocení, umožňují kontrolu procesu ze vzdáleného pracoviště, případně z mobilního zařízení a jsou schopny pokrýt geograficky rozlehlé oblasti. [3]

Podobnou roli plní také distribuované systémy řízení (DCS), které se vyznačují značnou robustností, redundancí a integrací velkého množství funkcí na jedné platformě. Často jsou využívány v odvětvích zpracovávajících produkty s vysokou přidanou hodnotou, či při práci v nebezpečném prostředí. Výrobci DCS často dodávají nejen samotný software, ale i nezbytné hardwarové vybavení, čímž minimalizují problémy s kompatibilitou jednotlivých zařízení při integraci. Novější platformy dnes již standardně zvládají pomocí otevřených komunikačních protokolů pracovat s hardwarem libovolného výrobce, disponují tenkým klientem přístupným z webového prostředí, nebo skrze mobilní aplikace a obsahují pokročilé systémy pro zpracování a vyhodnocení velkých objemů provozních dat.[4]

Následující text diplomové práce byl zasvěcen problematice supervizního řízení. Úvodem se věnuje současnému stavu techniky v oblasti supervizního řízení, je popsán vývoj HMI (Human Machine Interface – rozhraní člověk/stroj) a poskytnut přehled o současně nejpožívanějších softwarových řešeních pro řízení procesů a jejich vizualizaci. Následující kapitoly se věnují pojmům z oblasti informatiky, distribuovaných systémů a virtualizace.

Jedním z cílů práce je vytvoření demo úlohy reprezentující DCS. Na teoretický úvod proto navazuje popis současného stavu demo úlohy pro řízení asynchronního motoru realizovaného pomocí softwaru Compact Control Builder. Hardwarové vybavení použité pro současnou demo úlohu je následně zrevidováno a použito k tvorbě nové demo úlohy vytvořené za pomoci DCS systému 800xA od firmy ABB. Nová demo úloha je specifikována tak, aby co možná nejvíce přiblížila reálné průmyslové využití dostupných komponent.

Druhá významná část diplomové práce obsahuje funkční specifikaci nové demo úlohy včetně doplňujících informací k nově použitým komponentám propojených v distribuovaném systému řízení nainstalovaném na virtuálních počítačích. Dále je popsána firmwarové příprava použitého hardwaru následovaná rozborem vývojového diagramu demo úlohy. Další část diplomové práce se věnuje programování ve vývojovém prostředí pro tvorbu aplikací pro PLC a závěrem je rozebráno testování funkčnosti aplikace při myslitelných poruchových stavech.

Třetí část diplomové práce je věnována tvorbě vizualizačních obrazovek pro operátorské pracoviště a dotykový panel, pomocí kterých je demo úloha řízena. V příslušných kapitolách budou nejprve popsány základní principy práce s programy určenými pro tvorbu obrazovek a následně budou obrazovky na základě funkční specifikace vytvořeny. Poslední kapitola shrnuje důležité poznatky k bezpečnému provozu demo úlohy.

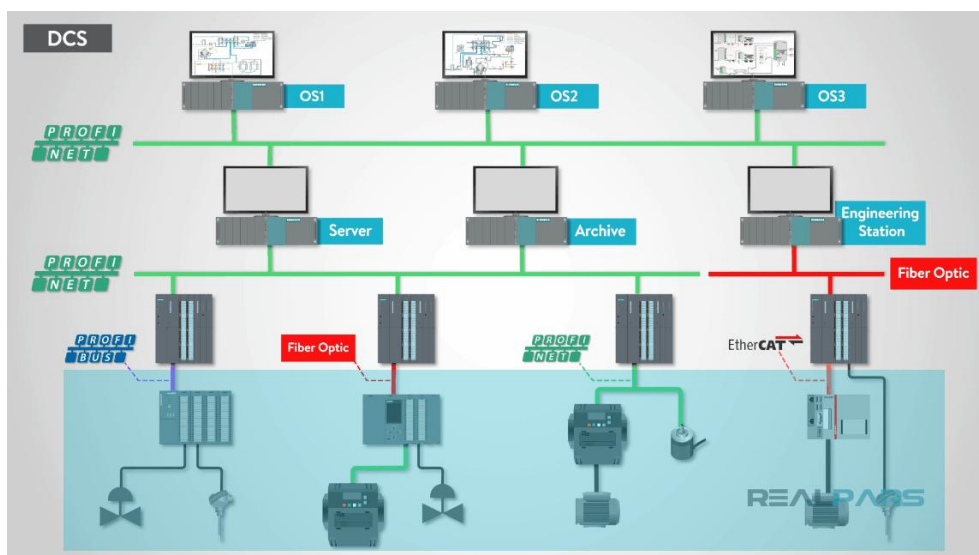
V závěru práce byly shrnuty a diskutovány dosažené výsledky a použité postupy a je navržen další směr vývoje práce.

1 Současný stav techniky supervizního řízení

Následující kapitola shrnuje současný stav techniky supervizního řízení, shrnuje vývoj HMI, popisuje důležité součásti SCADA systému a nabízí výčet nejpoužívanějších distribuovaných systémů řízení.

1.1 Distribuovaný systém řízení

DCS řízení představuje pokročilý systém schopný řídit složité výrobní procesy napříč celou továrnou. Na rozdíl od pouhého využití samostatných PLC ke kontrole jednotlivých procesů je s DCS možné sbírat a kontrolovat data z velkého množství kontrolérů a PLC. Jádrem systému není procesor v PLC, ale nadřazený počítač(server), který shromažďuje a vyhodnocuje data z jednotlivých kontrolérů. DCS je často využíván pro složité kontinuální procesy v oblasti chemie, zpracování ropy a potravinářství. Řízení DCS je většinou realizováno z velínů vybavených SCADA technologií, ale zároveň je dnes dostupné také ve webovém a mobilním prostředí.



Obrázek 1.1 Ukázka struktury DCS [5]

Z povahy vysoké přidané hodnoty zpracovávaných produktů je spolehlivost těchto systémů jednou z hlavních priorit. Velkou výhodou je využití distribuovaného modelu, kdy je možné zdvojit a zálohovat nejen přenosové vedení a databázové systémy, ale také řídicí aplikace. Značnou výhodou je také skutečnost, že při výpadku řídicích aplikací nedojde k nefunkčnosti jednotlivých kontrolérů a dalších zařízení, které budou i nadále pracovat a vykonávat řídicí smyčku. Pokud například přestane fungovat archivační server, nemá tento výpadek vliv na okamžitou funkčnost SCADA systému a aplikace je i nadále říditelná. Nejzranitelnější částí systému jsou databázové servery zajišťující zpracování dat a komunikaci mezi I/O moduly a dalšími částmi systému. Z toho důvodu bývají často zdvojené a pravidelně zálohované. Obecně jsou DCS vhodné pro aplikace s velkým

množstvím vstupů a výstupů, dosahujících v mnoha případech až řádů milionů. Distribuované systémy jsou řízeny pomocí HMI, rozšířených operátorských pracovišť, kolaborativních panelů, webových prohlížečů či operátorských stanic umístěných na osobních počítačích. [5]

1.2 Vývoj HMI

Základní úlohou HMI je tvorba rozhraní mezi člověkem a řídicím systémem. HMI si klade za úkol poskytnout operátorovi všechny informace potřebné pro řízení výroby, řešení alarmových stavů a v neposlední řadě sledování historie výrobního procesu či zpracování statistických trendů. Velký důraz při vývoji HMI je kladen zejména na přehlednost, čehož je dosahováno mimo jiné pokroky v zobrazovací technice.

HMI 1.0

Počátky vývoje HMI spadají do období zavádění elektronických a logických prvků do průmyslové výroby. S nástupem elektrických čidel a elektrických spínacích prvků bylo umožněno dálkové řízení procesů. Výstupy technologií bylo možné svést do centralizovaného velína, odkud byly jednotlivé procesy řízeny pomocí základních ovládacích prvků - tlačítek, páček, potenciometrů a dalších. Často byly využívány signalizační diody, různé budíky, segmentové displeje, tlačítka a potenciometry. [6]



Obrázek 1.2 Velín jaderné elektrárny[7]

HMI 2.0

Nástup druhé generace HMI je spojován s vývojem informačních technologií. Díky miniaturizaci a dostupnosti počítačů se otevřeli možnosti pro jejich implementování do výrobních procesů. S příchodem počítačové grafiky došlo k velkému zpřehlednění monitorovacích systémů a rozšířili se zejména možnosti názorné vizualizace. Lokální vizualizace běží nejčastěji na ovládacím panelu připojeném přímo ke stroji, nebo na osobním počítači v těsné blízkosti technologie. Díky možnostem vycházejícím z distribuovaného systému řízení a počítačovým sítím je možné také dálkové řízení procesů z velínů, kde počítače nahradili kontrolky a tlačítka. [6]

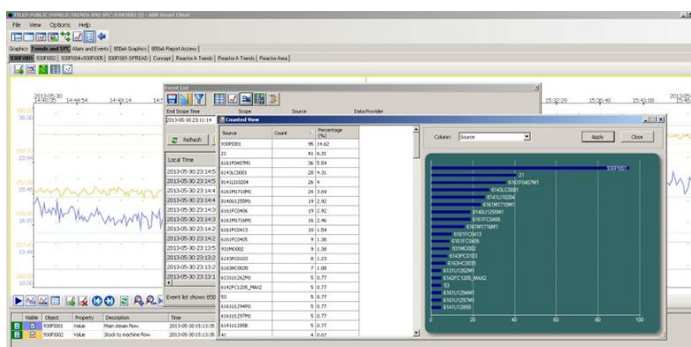


Obrázek 1.3 Vizualizační panel umístěný na stroji

HMI 3.0

S narůstající konektivitou a vývojem internetu je spojován další významný milník v oblasti HMI. Firemní, informační, řídicí a vizualizační systémy jsou postupně převáděny na webové aplikace, které jsou dostupné z privátní sítě nebo prostředí internetu. Výhodou převodu řídicích systémů na webové aplikace je nezávislost na hardwarovém i softwarovém vybavení a přístup do systému odkudkoliv na světě na základě ověření uživatelských práv. Podstatnou proměnou pak procházejí systémy, které přecházejí k architektuře orientované na služby (SOA) a využívají serverové nebo cloudové řešení. [4]

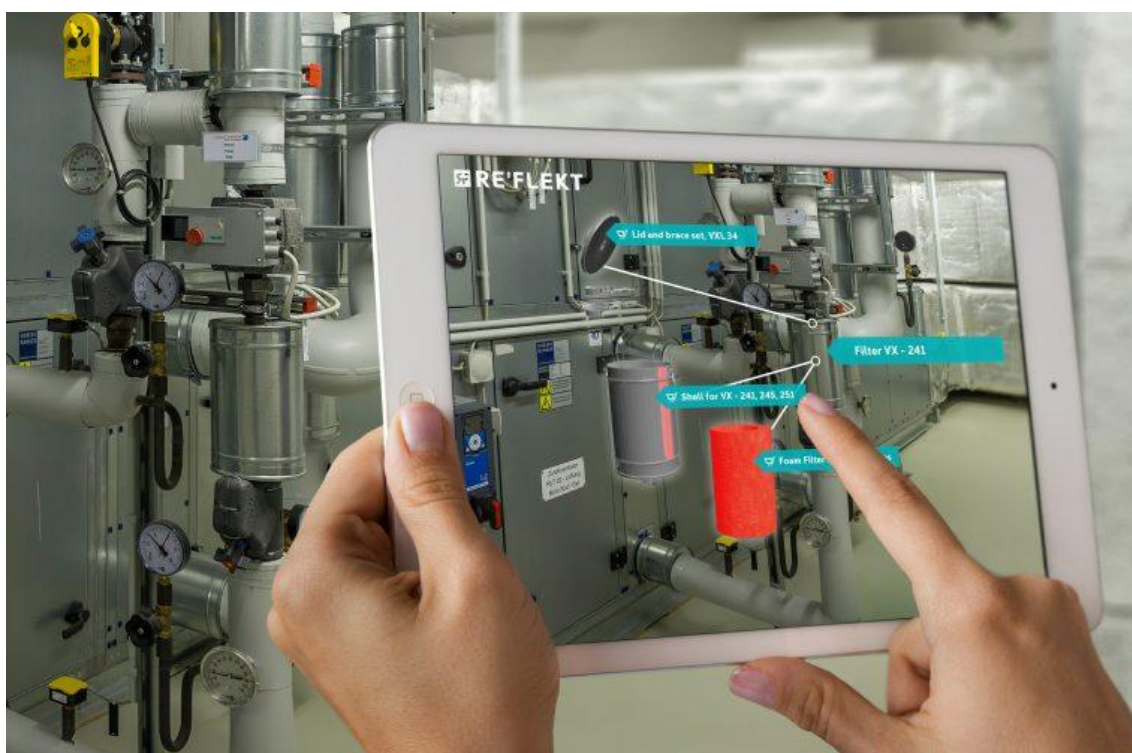
[6]



Obrázek 1.4 800xA Smart Client [4]

HMI 4.0

S rostoucím podílem síťových prvků umožňujících rychlé a kvalitní bezdrátové připojení začíná do oblasti HMI pronikat virtuální a augmentovaná realita. Díky augmentované realitě mohou být provozní data zobrazována přímo v místě umístění technologie bez nutnosti fyzické instalace zobrazovací techniky. Rozšířená realita může sloužit zejména ke zvýšení přehlednosti a informovanosti obsluhy či údržby. Virtuální realita nabízí velké možnosti zejména v oblasti návrhu a testování systémů, tréninku a výuce. Pro možnost skutečného řízení strojních zařízení za využití rozšířené reality je nutno zobrazovací technologie doplnit také technologiemi umožňujícími záznam podnětů operátora. To může být dosaženo například pomocí technologií schopných rozeznat gesta či hlasové pokyny, instalovaných přímo ve výrobních halách. [6][8]



Obrázek 1.5 Rozšířená realita realizovaná na chytrém zařízení[8]

1.3 Základní funkce SCADA systémů

Následující podkapitola obsahuje základní požadavky kladené na aplikace označované jako SCADA systémy.

Sběr Dat

Data z PLC jsou pomocí middleware rozhraní zasílány na databázové servery, odkud k nim přistupují další zařízení v síti, včetně řídicího počítače. Většina moderních SCADA systémů dokáže číst data z libovolných kontrolérů díky podpoře komunikačního protokolu OPC (OLE for Process Control).[9] [10][11]

Vizualizace a kontrola

Hlavním sdělovacím prostředek SCADA systémů jsou vizualizační okna obsahující informativní grafiku doplněnou o ovládací prvky. Okna jsou přizpůsobitelná uživatelským aplikacím a je u nich kladen velký důraz na přehlednost. Informativní prvky mají často podobu budíků, kontrolek, stavoznaků a dalších měřících přístrojů převedených do digitální podoby. Kromě jiného bývá součástí oken také tabulka událostí, která společně s časem a dalšími volitelnými údaji zaznamenává důležité změny v aplikaci, starty podsystémů, alarmy a další. Průběh sledovaných veličin je často zobrazován v grafu. Ovládacími prvky jsou různá tlačítka, přepínače, slidery nebo klávesnice. Prvky jsou realizovány pomocí objektově orientovaného programování, je jim možné přidělit grafickou podobu a propojit je s I/O tagy. [12] [10][11]



Obrázek 1.6 Příklad vizualizační obrazovky v programu Wonderware InTouch[13]

Alarmy

Systém je na základě stanovených limitů schopen varovat operátora o nastalé mimořádné situaci a zároveň by měl nabídnout vhodná řešení. Alarmové stavy mohou nabývat třech základní hodnot. První je nepotvrzený alarm (operátor jej zatím nevzal na vědomí), potvrzený alarm (operátor vzal alarm na vědomí) a bezalarmový stav (alarm skončil návratem do normálu). Alarmy se často projevují blikáním, změnou barvy nebo novou položkou v logu událostí. [10][11]

Alarmy – distribuované 16:25:39 1. 8. 2013

Datum a čas	Stav	Typ	Priorita	Jméno	Skupina	Hodnota	Poskytovatel	Limit
01.08/2013 04:21:42 odp.	UNACK	HI	5	APVEBT	COV	90	\ntouch	85
01.08/2013 04:21:42 odp.	UNACK	DSC	1	IOStatusModicom	Modicom	OFF	\ntouch	OFF
01.08/2013 04:21:42 odp.	UNACK	DSC	1	StatusModicom	Modicom	OFF	\ntouch	OFF
01.08/2013 04:21:45 odp.	UNACK_RTN	LOLO	1	USERFree	\$System	50	\ntouch	30
01.08/2013 04:21:45 odp.	UNACK_RTN	LOLO	1	GDIFree	\$System	50	\ntouch	30
01.08/2013 04:22:00 odp.	UNACK_RTN	LOLO	1	DiskFree	\$System	185	\ntouch	5
01.08/2013 04:24:24 odp.	UNACK_RTN	HI	10	ProdLevel	Reactor	3927	\ntouch	4000
01.08/2013 04:25:04 odp.	ACK	DSC	1	M1	\$System	OFF	\ntouch	OFF
01.08/2013 04:25:13 odp.	UNACK	HI	5	ReactLevel	Reactor	1840	\ntouch	1800
01.08/2013 04:25:29 odp.	UNACK	HI	5	ReactTemp	Reactor	100.9	\ntouch	100
01.08/2013 04:25:29 odp.	UNACK_RTN	HI	5	ROLevel	COV	85.2	\ntouch	90

Displaying 1 to 11 of 11 alarms. Default Query 100 % Complete

Potvrď všechny Potvrď vybrané Potvrď zobrazené Potvrď poslední Potlačit vybrané Potlačit skupinu Potlačit prioritu

Kontextové menu Nastavení třídění Nastavení filtru Návoděda

Obsah vybraných sloupců alarmového objektu
08 | 16:21:57 | UNACK | HI | 5 | APVEBT | COV | \ntouch | 90 / 85 | None

Počet alarmů 11 Statistika přenosu

Typ: ☒ Současné ☐ Historické

Stav: ☒ Všechny ☐ Potvrzené ☐ Nepotvrzené

Priorita: Od: 1 Do: 999

Skupina: Reaktor Reaktor Reaktor Všechny

Obrázek 1.7 Záznam alarmů z aplikace Wonderware Historian[13]

Trendy

Většina prostředí umožňuje prohlížet a vyhodnocovat historická data, na jejichž základě je možné optimalizovat řídicí aplikace nebo revidovat alarmové stavy. Data je možné zpracovat přímo v aplikaci, nebo je exportovat k další analýze externími nástroji. Dalšími důležitými funkcemi je filtrování vybraných hodnot nebo zpřehlednění pomocí grafického výstupu. [10][11]

Zabezpečení

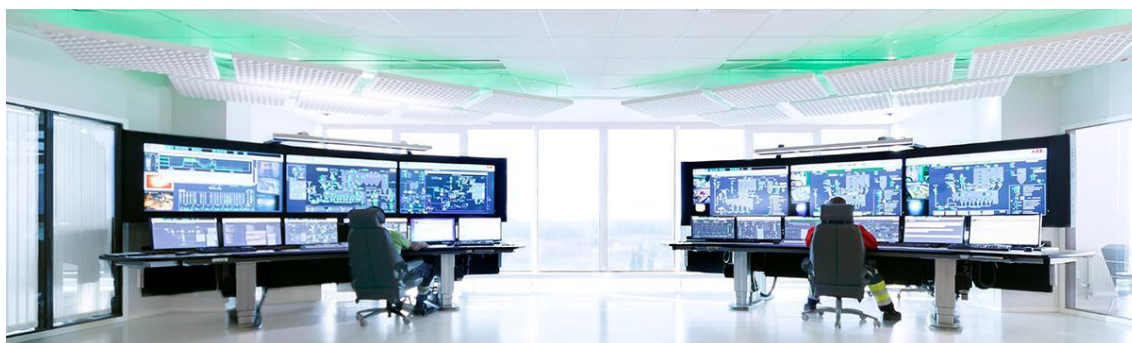
Z důvodu možnosti přihlášení skrze webové prostředí i mobilní aplikace je velký důraz kladen také na zabezpečení SCADA systémů. Základní funkcí je možnost zabezpečeného přihlášení s různým stupněm oprávnění. Tento přístup umožňuje rozlišovat jednotlivé uživatele přihlášené k aplikaci a následně jim otevřít předdefinované okna na základě stupně oprávnění. [10][11]

1.4 Přehled dostupných SCADA a DCS řešení

Současný trh nabízí rozličné možnosti v oblasti DCS a SCADA řízení, přičemž při volbě konkrétního řídicího systému vždy záleží zejména na požadavcích na funkci a také na finančních možnostech zákazníka. V případě velkých provozů je z pohledu integrace a robustnosti výhodnější využít distribuovaných systémů řízení, kdežto v případě menšího počtu řídicích smyček a I/O je často finančně výhodnější přiklonit se ke SCADA řízení. Kromě níže zmíněných softwarových řešení jsou využívány také následující programové prostředky: Yokogawa Centrum CS3000, Honeywell TDC 3000, Emerson Delta V, LabView, Promotic či Allen-Bradley NetLinx. [14]

ABB 800xA

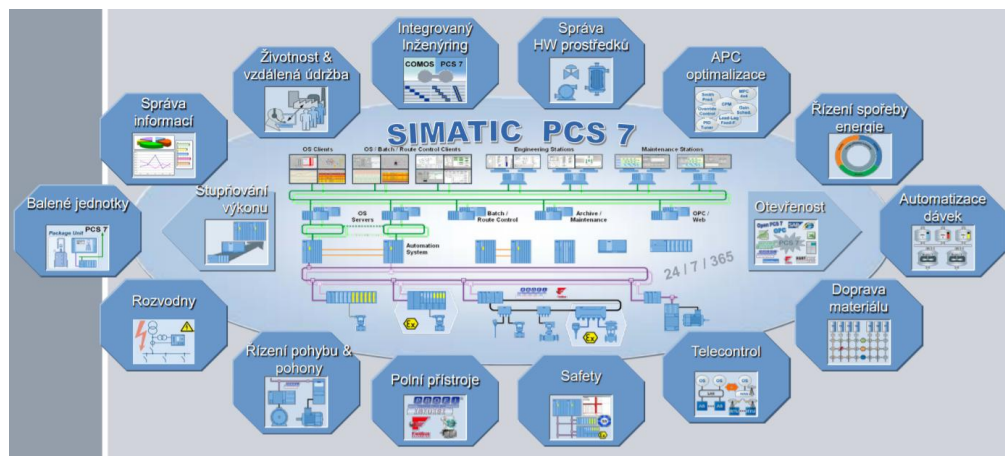
Systém ABB 800xA je jedním z nejvyžívanějších DCS s více než 10 000 průmyslovými instalacemi. Mezi jeho hlavní přednosti patří spolehlivost, robustnost a bezpečnost klient - server systému běžícím na průmyslové počítačové síti. Jako jedno z prvních průmyslových řešení od ABB byl systém 800xA propojen s rozhraním ABB Ability – unifikovaným, multiprůmyslovým rozhraním propojujícím zařízení a data na všech úrovních, od field devices po cloud. ABB Ability představuje řešení pro průmyslový internet věcí propojený s cloudovou infrastrukturou. Mezi výhody platformy patří zejména důraz na zabezpečení a správu dat. Propojení ABB Ability s DCS 800xA přináší nový vhled do správy průmyslových podniků a díky orientaci na implementaci VR a umělé inteligence (AI – Artificial Intelligence) bude v budoucnu jedním z důležitých hráčů na poli distribuovaných řídicích systémů. Systému 800xA je dále věnována samostatná podkapitola.[15]



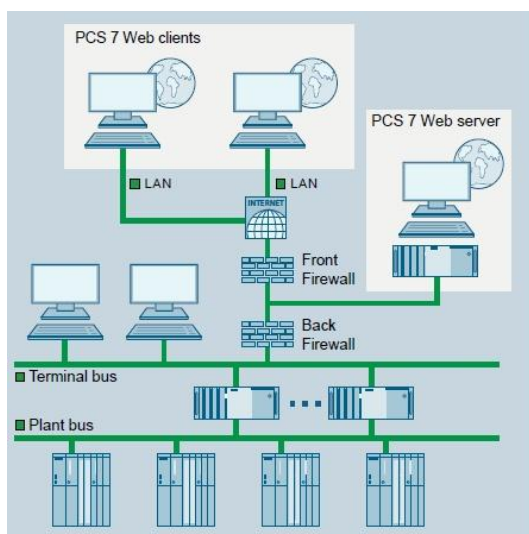
Obrázek 1.8 Rozšířené operátorské pracoviště s ABB 800xA[16]

Siemens Simatic PCS7

Siemens PCS7 je integrované a homogenní automatizační DCS řešení pro všechny odvětví průmyslu s jednotnou databází objektů a centrální správou projektu. Hardwarově a softwarově je realizován za použití komponent z řady SIMATIC doplněných o integrované systémové služby jako diagnostiku, alarm management, asset management a další. Systém je vhodný pro malé systémy od cca 100 I/O až po systémy velikosti stovek tisíc I/O. Kromě tradičních PLC je možné využít také periferie vhodné pro DCS. Důležitá je také implementace nadstavby SIMATIC Batch určené pro řízení dávkové výroby. Díky využití Advanced Process Control (APC) je možné pro řízení složitých procesů využívat nové technologie jako soft senzory, neuronové sítě a fuzzy řízení. Samozřejmostí je také možnost sdílet živá data do dalších aplikací jako např. MS Excel. Nechybí ani možnost vzdáleného přístupu přes webové rozhraní. Mezi zajímavé funkcionality systému patří možnost sledování stavu mechanických komponent ve výrobě, energetický management pro řízení spotřeby energií či nadstavba pro monitorování dopravy materiálu a látek.[17]



Obrázek 1.9 Přehled funkcionalit DCS Siemens PCS7[17]



Obrázek 1.10 Ukázka architektury PCS7

Wonderware InTouch

Wonderware InTouch je software pro vizualizaci, sběr dat a supervizní řízení technologických procesů kategorie SCADA/HMI s vysokou uživatelskou přívětivostí a celosvětovým rozšířením. InTouch umožňuje operátorům, technologům, kontrolorům i manažerům v reálném čase sledovat a reagovat na průběhy veškerých výrobních operací prostřednictvím názorného grafického znázornění libovolných technologických procesů. Nově pracuje s pojmem „Situační povědomí“ což je způsob tvorby obrazovek SCADA systémů, který umožňuje operátorům soustředit se na obsah, odstraňuje rušivé prvky, a tím snižuje únavu lidské obsluhy i při dlouhých pracovních směnách. Výsledným efektem je hladký chod sledovaného zařízení, méně prostojů a odstávek. InTouch je možné spustit z jakéhokoliv zařízení vybaveného webovým prohlížečem – může to být počítač, ale také tablet či chytrý telefon. Aplikaci InTouch je možné otevřít také z jiného místa podniku pomocí vnitropodnikové sítě nebo s využitím internetu odkudkoliv na světě. Klíčovou vlastností HMI/SCADA systému je schopnost propojení zařízení od mnoha různých výrobců, používajících pro předávání hodnot různé protokoly. Celosvětově nejrozšířenější protokoly a PLC jsou pokryty DAServery dodávanými přímo společností Wonderware. Samozřejmostí je podpora protokolů Modbus a OPC, které patří k základním standardům v oblasti průmyslové automatizace. Pro potřeby připojení k méně obvyklým zařízením nabízí program knihovnu Wonderware DAServer toolkit, která vývojářům umožňuje napsat vlastní DAServer na míru podle potřeb koncového zákazníka.[18]



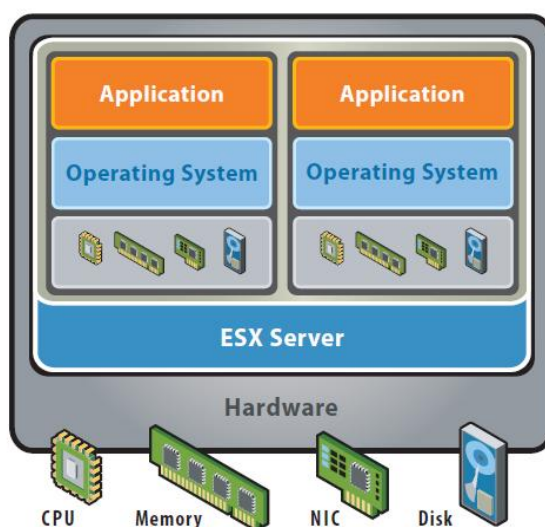
Obrázek 1.11 Wonderware InTouch[18]

1.5 Virtualizace

Pojem virtualizace označuje techniky používané pro vytvoření abstraktní hardwarové vrstvy, která umožňuje fyzické části počítače jako je paměť, procesor, úložiště a další, virtuálně rozdělit na několik samostatných zařízení. Vytvořené zařízení se nazývají virtuální stroje (Virtual Machines – VMs) a používají vlastní operační systém. Díky tomu se chovají jako nezávislé počítače, které dokážou využívat přidělené hardwarové prostředky. Mezi další benefity, které virtualizace přináší, patří možnost bezpečného testování aplikací či přímo celých operačních systémů v prostředí, které může být naprosto odděleno od okolí, vývoj softwaru pro fyzicky nedostupný hardware, možnost propojení diskových jednotek, rozsáhlé cloudové operace a další. [19][20]

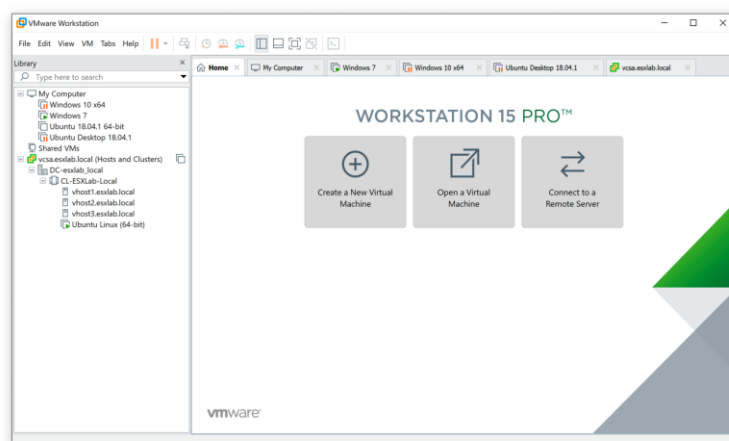
Virtuální stroj představuje simulaci fyzického počítače v softwarové podobě a obvykle se skládá z jednoho nebo více souborů, které představují samotnou konfiguraci VM, úložiště a tzv. snapshoty, neboli obrazy systému v daném čase, které i po havárii virtuálního stroje umožňují snadnou obnovu do provozuschopného stavu.[19]

Aplikace starající se o provoz jednotlivých virtuálních strojů na fyzickém počítači se nazývá Hypervisor. Tento ovládací software má za úkol spravovat výpočetní zdroje fyzického počítače a přidělovat je jednotlivým VMs dle předem daného schématu. Dle způsobu přístupu k hardwaru můžeme Hypervisory rozdělit do dvou kategorií. První kategorie obsahuje Hypervisory často používané u serverů, kdy v podstatě nahrazují klasické operační systémy a přímo přistupují k fyzickým zdrojům. Druhou kategorií jsou pak softwarová řešení běžící v prostředí operačního systému fyzického počítače. To znamená, že Hypervisor přistupuje k hardwaru skrze operační systém fyzického počítače.[19]



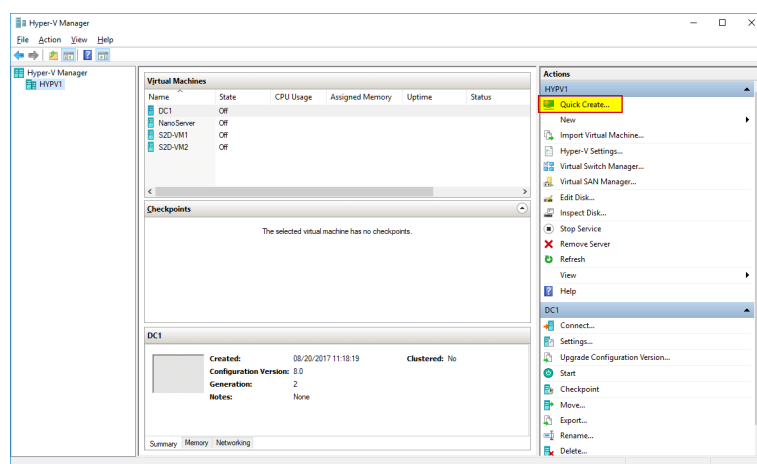
Obrázek 1.12 Virtualizace s ESX serverem[21]

Mezi oblíbené virtualizační platformy patří například software Worskatiton a ESX Server od americké firmy VMware. Aplikace je určena pro operační systémy Windows a Linux a v bezplatné verzi nabízí běh jednoho virtuálního stroje. Minimální požadavky pro zprovoznění programu jsou 64bitový procesor s minimálním taktem 1GHz, 2GB operační paměti a 150 MB místa na disku. Mezi přednosti programu patří implementace některých komerčně úspěšných rozhraní jako DirectX. Dále pak uživatelsky přívětivý přenos souborů mezi virtuálním a fyzickým počítačem a v neposlední řadě také implementace snapshotů.



Obrázek 1.13 Prostředí VMware Workstation 15 PRO[22]

Často využívaným softwarem je také Hyper-V od firmy Microsoft, který je vhodný zejména pro vizualizaci serverů a operačních systémů Windows. Z hlediska systémových a hardwarových požadavků stojí za povšimnutí, že aplikaci není možné nainstalovat do OS Windows 10 Home, Mobile ani Mobile Enterprise, vyžaduje 64bitový procesor a 4GB paměti RAM. K výhodám doazajista patří dobře zvládnutá síťová konektivita pro práci s fyzickými a virtuálními sítěmi. [23]



Obrázek 1.14 Virtualizační program Hyper-V[23]

2 Popis původního modelového prostředí demo úloh

V následující kapitole byl popsán výchozí stav hardwaru a softwaru dodaného firmou ABB pro potřeby výuky a prezentace dostupných technologií v rámci výuky.

2.1 Popis úloh

Obě současné demo úlohy, řízení asynchronního motoru a regulace poruchových veličin, vychází z dostupného hardwarového vybavení učebny automatizace, které zahrnuje asynchronní motor, frekvenční měnič, PLC s digitálními a analogovými kartami, rozvaděč s potenciometry a osobní počítače. Uvedená sestava sloužila jako demonstrátor řízení motoru a pokusný hardware pro různé realizace řízení a regulace pomocí PLC a SCADA systémů. V rámci modernizace technického vybavení učeben bylo přistoupeno k výměně počítačů a upgradu operačního systému. Tyto změny způsobily nedostupnost většiny původních úloh, funkční zůstala pouze základní úloha pro řízení motoru s jednoduchou vizualizací v programu Compact Control Builder. Skrze vizualizační okno bylo možné motor spustit příkazem Drive Start, zastavit příkazem Drive Stop, případně resetovat nastavení příkazem Drive Reset a dále bylo možné nastavovat požadované otáčky či požadovaný moment. Mezi zobrazovanými informacemi byla k nalezení aktuální rychlost, proud, moment a napětí. Další úlohy nebyly v nové sestavě dostupné. Mimo nově vzniklou místní síť zůstal počítač s historickou verzí softwaru 800xA, jehož zapojení do nového prostředí na základě nedostatečné hardwarové a softwarové konfigurace nemělo význam.[24],[25]



Obrázek 2.1 Přehled současného stavu pracoviště

2.2 Hardwarové konfigurace

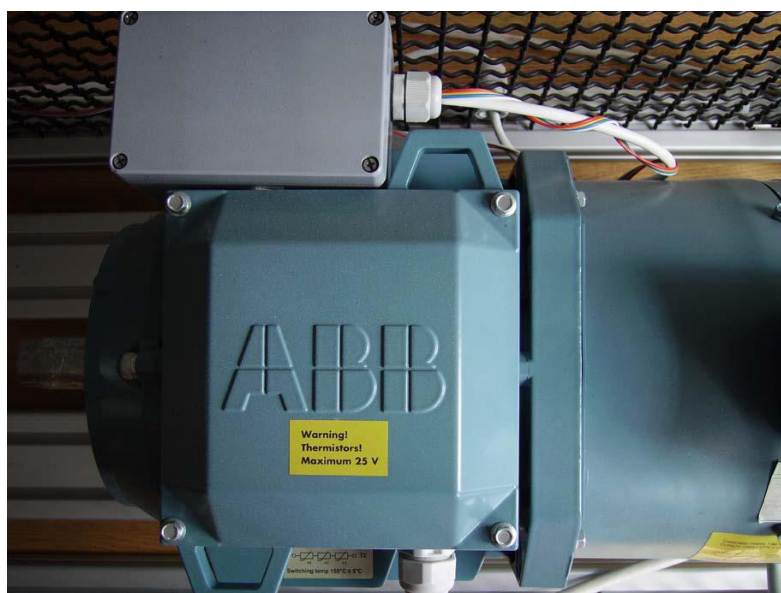
Následující podkapitola obsahuje výčet hardwarových komponent využívaný v původních demo úlohách a také parametry použitých počítačů.

Motor s příslušenstvím

Asynchronní motor M2AA 112M s kotvou nakrátko od firmy ABB je umístěn v pletivové kleci zajišťující oddělení prostoru výstupní hřídele od okolí. Řízení motoru je realizováno frekvenčním měničem. K motoru je připojeno přídavné chlazení Kurt-Maier pro udržování vyhovující pracovní teploty motoru při režimech vysoké zátěže. Monitorování teploty je řešeno pomocí PTC termistoru připojenému k ochrannému relé. Monitorování otáček má na starosti inkrementální snímač otáček. [24],[16]

Tabulka 2.1 Vybrané parametry motoru ABB [24],[16]

Parametr	Hodnota
Jmenovité napětí	400 V
Jmenovitý proud	7,2 A
Jmenovitá frekvence	50 Hz
Jmenovité otáčky	2850 ot/min
Jmenovitý výkon	4 kW
Krytí IP	55



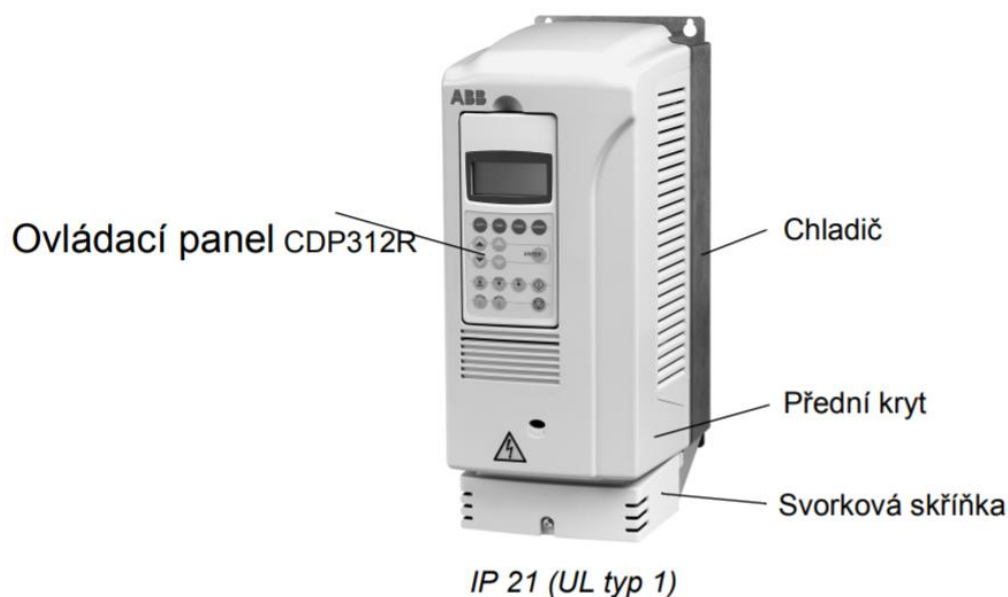
Obrázek 2.2 Motor ABB M2AA 112M s přídavným chlazením a snímačem otáček[24]

Frekvenční měnič

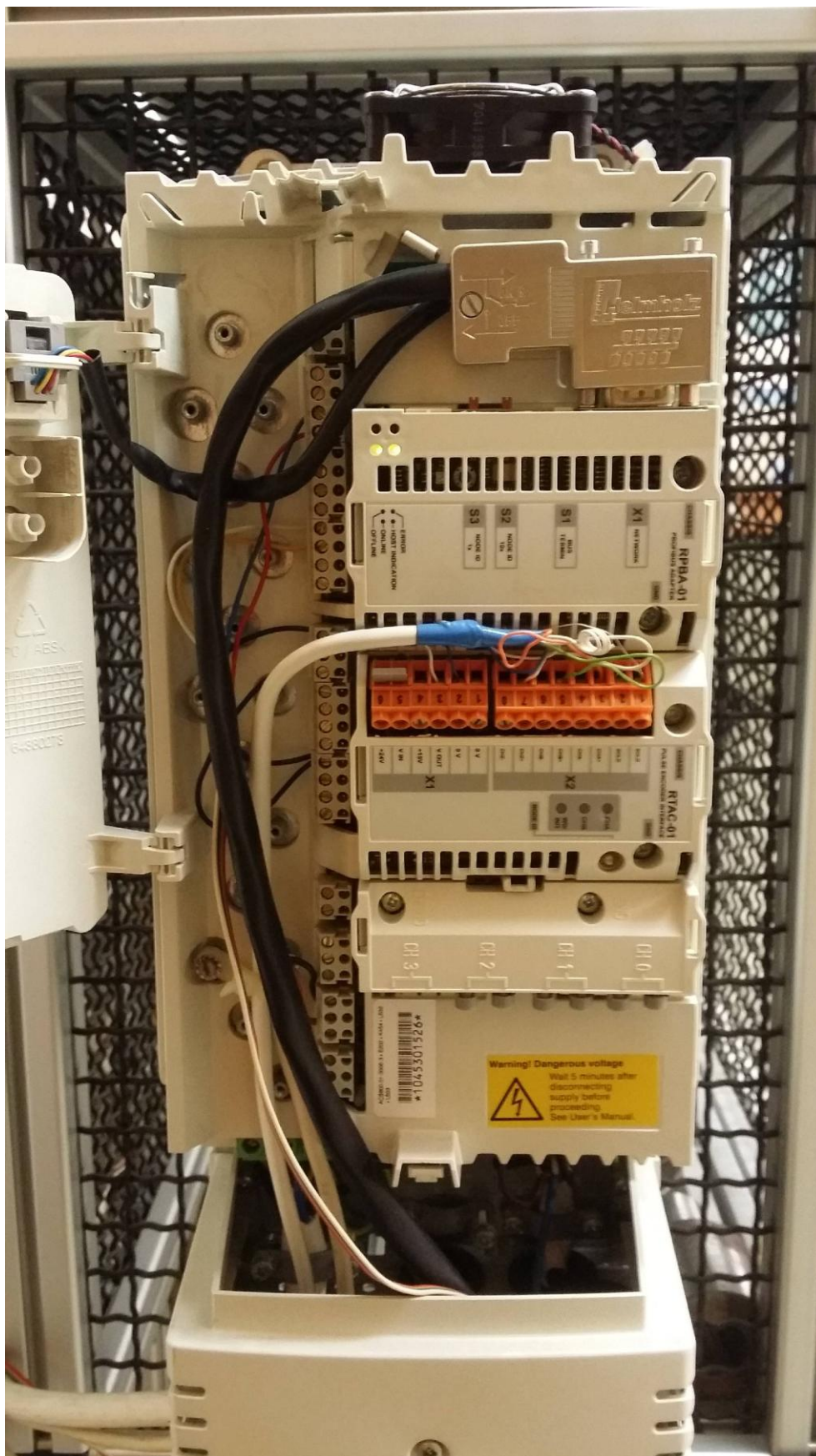
Řízení motoru je realizováno pomocí frekvenčního měniče ABB ACS 800-01-0006. Měnič lze řídit skalárně, nebo za pomoci technologie DTC (Direct Torque Control). Zařízení je ovladatelné odnímatelným panelem CDP 312R na čelní straně, nebo použitím externího řízení a síťové komunikace realizované příslušným modulem RPBA připojeným k PLC přes rozhraní PROFIBUS. Druhý použitý rozšiřující modul RTAC slouží k vyhodnocení pulzů ze snímače otáček. [16],[24],[26]

Tabulka 2.2 Vybrané parametry frekvenčního měniče

Parametr	Hodnota
Normalizovaný proud	10 A
Výstupní výkon max	10,9 kW
Počet fází	3
Vstupní napětí	380 – 415 V
Frekvence vstupního napětí	48 – 63 Hz
Váha	9 Kg
Krytí IP	21



Obrázek 2.3 Frekvenční měnič ABB ACS800 [16],[26]



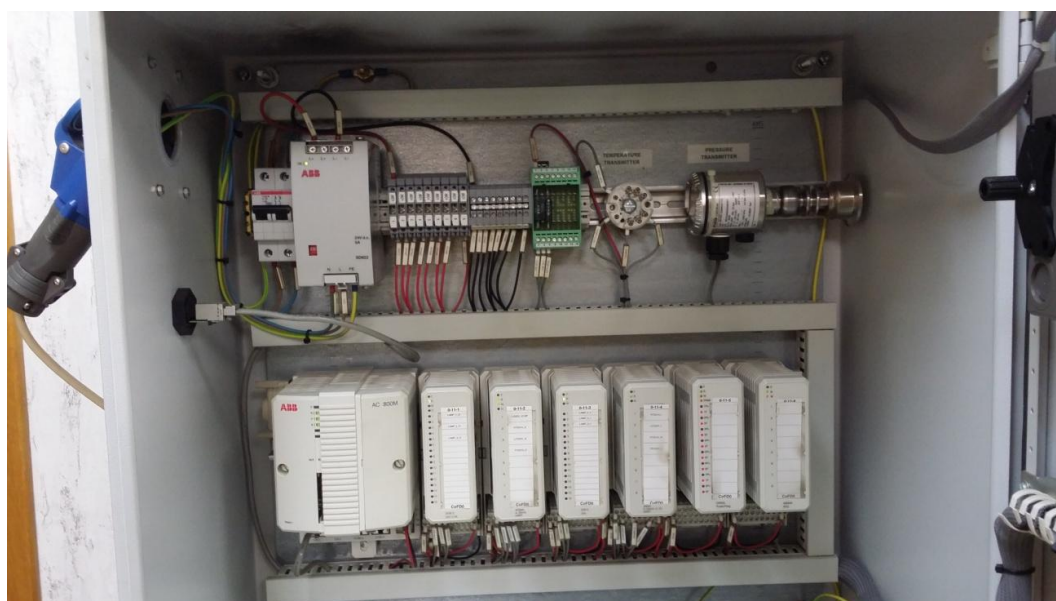
Obrázek 2.4 Moduly frekvenčního měniče

Rozvaděč osazený tlačítky a potenciometry

Pro zadávání analogových a binárních vstupů slouží potenciometry a tlačítka umístěná na dveřích rozvaděče. V minulosti sloužila konfigurace jako vstup pro regulační úlohy, s čímž souvisí také výstupní panel ABB SM500F pro kontinuální zobrazování analogových hodnot. Uvnitř rozvaděče je umístěno PLC s analogovými a digitálními vstupními i výstupními kartami, tlakový snímač a teplotní můstek. Za povšimnutí stojí propojení potenciometrů s tlačítky osazené světelnými diodami, kdy tlačítka v zamáčknutém stavu fyzicky rozpojují svorky analogového signálu a nulují jeho hodnotu.



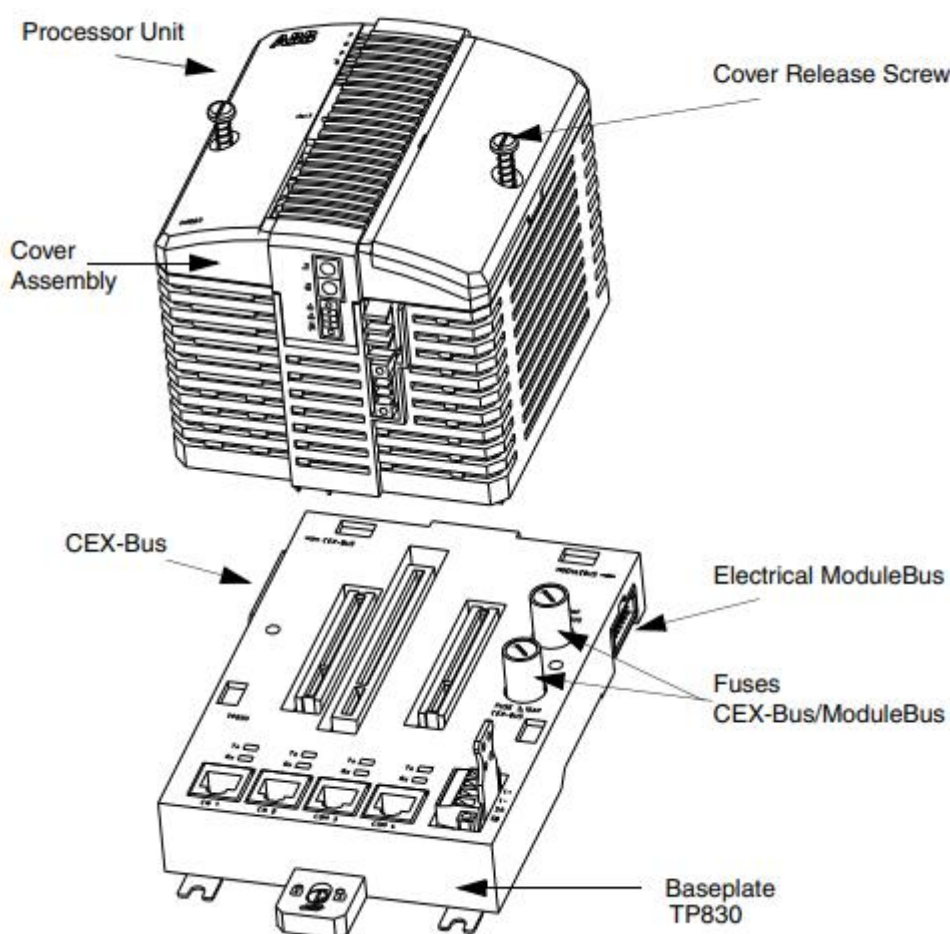
Obrázek 2.5 Dveře rozvaděče osazené tlačítky, potenciometry a zobrazovacím panelem



Obrázek 2.6 Rozvaděč s PLC 2

PLC ABB AC800M

Pro řízení frekvenčního měniče a následně také motoru, stejně jako pro přístup k tlačítkům a hodnotám potenciometru je využíváno modulární PLC AC 800M od firmy ABB. Díky značné rozšiřitelnosti a velkému množství volitelných I/O a komunikačních modulů je možné využívat PLC pro malé aplikace, ale také pro rozsáhlé projekty. Jednotlivé moduly jsou umísťovány na DIN lištu a propojeny přes sběrnici CEX. Po překročení povoleného množství modulů na sběrnici je možné připojit další ostrov modulů pomocí rozhraní ModuleBus nebo ProfiBus. Komunikace s nadřazenými systémy je možná díky rozhraní Ethernet a RS232. Základní procesorová jednotka je dodávána v různých provedeních frekvence procesoru a vnitřní paměti.[24],[27]



Obrázek 2.8 Procesorová jednotka a základna s komunikačním rozhraním[24],[27]

Vybrané parametry použitých PLC

Následující tabulky shrnují vybrané parametry použitých PLC ABB AC800M. Z obrázků je patrné umístění jednotlivých PLC a také propojení jejich modulů.

Tabulka 2.3 PLC 1 (Motor)

Parametr	Hodnota
Procesorová jednotka	PM860
Procesor	MPC860 48MHz
Operační paměť	2 MB flash PROM (firmware) + 8 MB SDRAN
Připojovací porty	CN1 + CN2 (Ethernet), COM3 (RS-232C), COM4 (RS-232C)
Krytí	IP 20
Rozměry	119x186x135
Interní záložní baterie	provoz po dobu 48 hodin.
Připojené moduly	DI801, DO801, AI895, AI830, RPBA-01

Tabulka 2.4 PLC 2 (Rozvaděč)

Parametr	Hodnota
Procesorová jednotka	PM851
Procesor	MPC860 48MHz
Operační paměť	2 MB flash PROM (firmware) + 8 MB SDRAN
Připojovací porty	CN1 + CN2 (Ethernet), COM3 (RS-232C), COM4 (RS-232C)
Krytí	IP 20
Rozměry	119x186x135
Interní záložní baterie	provoz po dobu 48 hodin.
Připojené moduly	DO810, AO845, DI810, AI845, DP840, AI830A



Obrázek 2.7 PLC 1 určené pro řízení motoru



Obrázek 2.8 PLC 2 umístěné v rozvaděči

Počítače

Pro správu a řízení výše popsaného hardwaru jsou využívány dva osobní počítače propojené v lokální síti. Komunikace s PLC probíhá po ethernetovém rozhraní. Následující tabulky uvádějí vybrané parametry obou počítačů.

Tabulka 2.5 Specifikace PC 1 (DESKTOP-DE801DT)

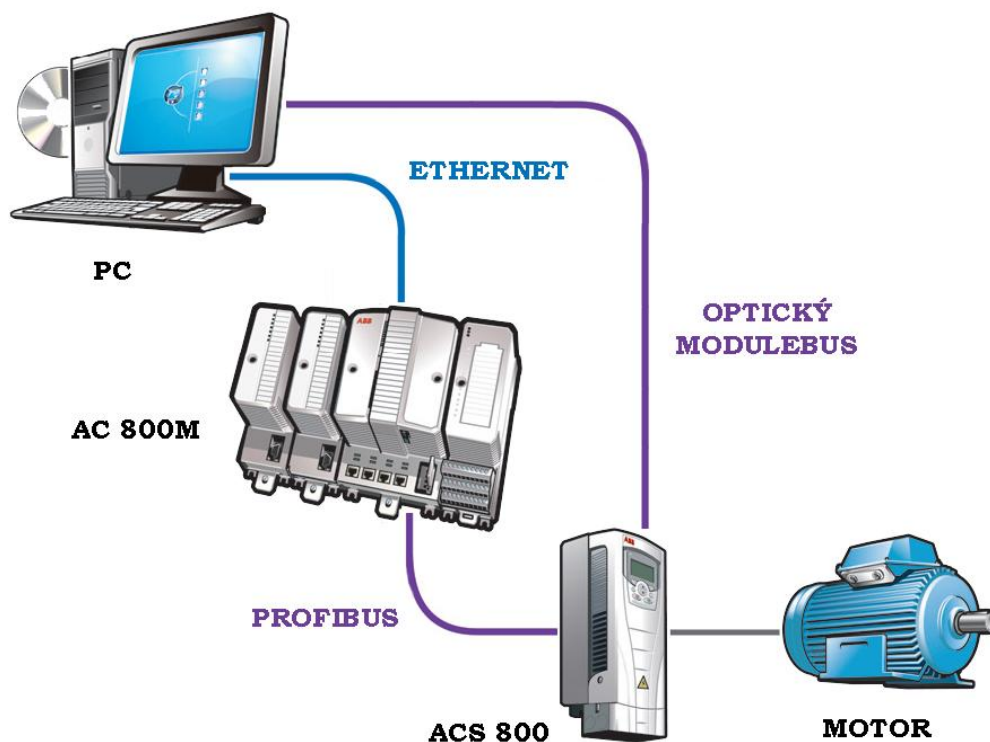
Parametr	Hodnota
Operační systém	Windows 10 Education 64 bit
Procesor	Intel Core i7-2600 3,4GHz
Operační paměť	16 GB
Grafická karta	Intel HD Graphics (Integrovaná)
Disková jednotka	ADATA SU800(SSD 512GB), TOSHIBA HDWD110 (HDD 1TB)
Síťové adaptéry	Intel 82579LM, Realtek PCIe GbE, Realtek RTL8139/810x
Další	USB 3.0

Tabulka 2.6 Specifikace PC 2(Flash_Dell)

Parametr	Hodnota
Operační systém	Windows 10 Education 64 bit
Procesor	Intel Core i7-2600 3,4GHz
Operační paměť	16 GB
Grafická karta	AMD Radeon HD6450
Disková jednotka	ADATA SU800(SSD 512GB), Seagate Barracuda 7200.12 (HDD 1TB)
Síťové adaptéry	Intel 82579LM, Realtek PCIe GbE, Realtek RTL8139/810x
Další	USB 3.0

Propojení PC1 a PLC 1 s frekvenčním měničem a motorem

Počítač PC_1 je připojen k PLC_1 pomocí ethernetové sítě a slouží k programování PLC a řízení motoru skrze frekvenční měnič. Frekvenční měnič je možné k počítači připojit přímo pomocí optického ModuleBusu, ale v současné aplikaci a v průmyslové praxi je nejčastěji používáno připojení k frekvenčnímu měniči skrze PLC. Obě možnosti připojení jsou znázorněny na obrázku níže. [24]



Obrázek 2.9 Možnosti připojení k frekvenčnímu měniči

2.3 Využívaný software

Následující podkapitola shrnuje původní softwarové řešení demo úloh, jejichž části mohou být využity při další práci na integraci nového řešení a nových demo úloh.

Windows

Oba počítače běží na 64-bitovém operačním systému Windows 10 Education umožňující práci s virtuálními počítači v programu HyperV.

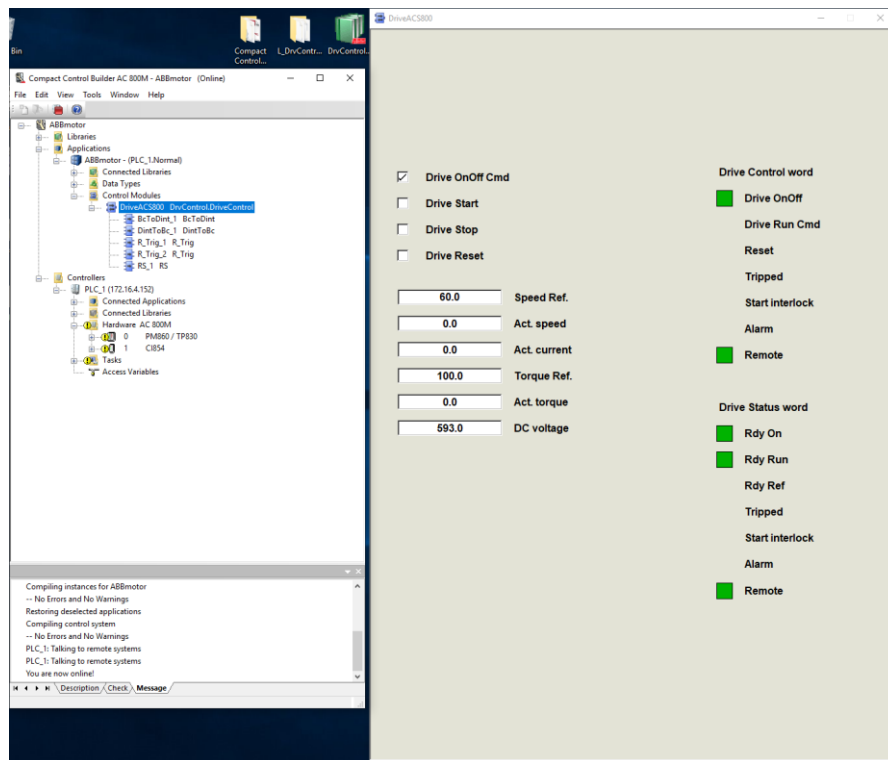
Compact Control Builder AC800M

Program Compact Control Builder je vývojovým nástrojem pro tvorbu PLC programů určených pro PLC AC800M. Na základě normy IEC 61131-3 podporuje následující standardní programovací jazyky. Ladder diagram (LD) určený pro tvorbu programů na základě reléových schémat, Function Block Diagram (FBD) vycházející z logických prvků, Instruction List (IL) programovací jazyk podobný assembleru, dále Structured Text (ST) představující vyšší programovací jazyk a v neposlední řadě Sequential Function Chart (SFC) určený pro grafické programování. [24],[28]

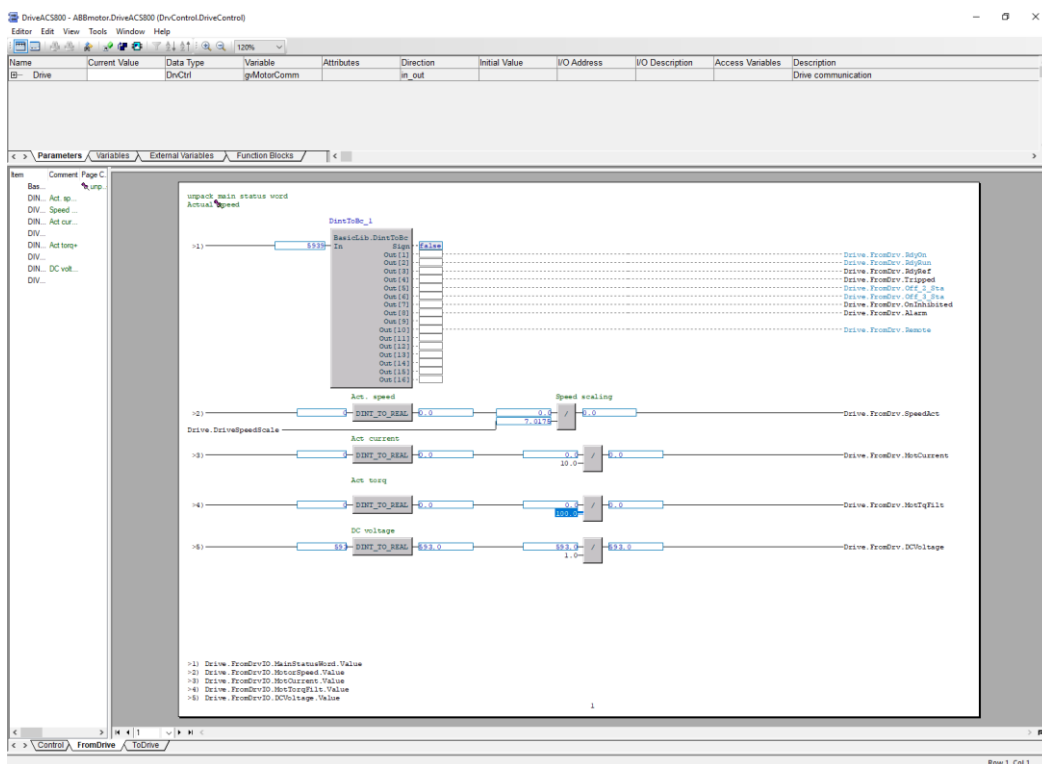
Hlavními funkcionalitami vývojového prostředí je integrace všech potřebných součástí pro vývoj aplikací, stejně jako knihoven ovladačů hardwarových součástí a široké využití simulačního prostředí pro důkladné testování aplikací.[28]

Jediným zachovalým nástrojem pro ovládání asynchronního motoru byl jednoduchý PLC program s názvem ABBmotor vytvořený v prostředí Compact Control Builder. Program byl vytvořen v rámci diplomové práce s názvem „Systém monitorování a řízení technologie“ inženýra Jaromíra Zavadila. Zvolený program se při testování podařilo zkompileovat a nahrát do PLC. Po přechodu do online režimu bylo motor možné ovládat pomocí jednoduchého vizualizačního panelu nabízejícího údaje o nastavené rychlosti, aktuální rychlosti, aktuálně odebíraném proudu, požadovaném momentu, aktuálním momentu a napětí. Uvádí také stavy kontrolních slov pro řízení motoru. Samotný program bylo možné využít pro další práci na nové demo úloze. Obrázky níže ukazují vizualizační panel a část programu psaného v jazyce FD.[24]

Pro práci s PLC 2 osazeném v rozvaděči se nedochoval žádný použitelný software. Na PLC byly v minulosti realizovány celkem dvě úlohy sloužící k simulaci výrobních procesů. První se věnovala problematice KPI(Klíčové ukazatele výkonnosti), druhá řízení průtoku kapalin. Pro potřeby nové demo úlohy nebyly původní úlohy zapotřebí, pro daný hardware byla vypracována nová úloha.



Obrázek 2.10 Jednoduchý vizualizační panel pro motor [24]



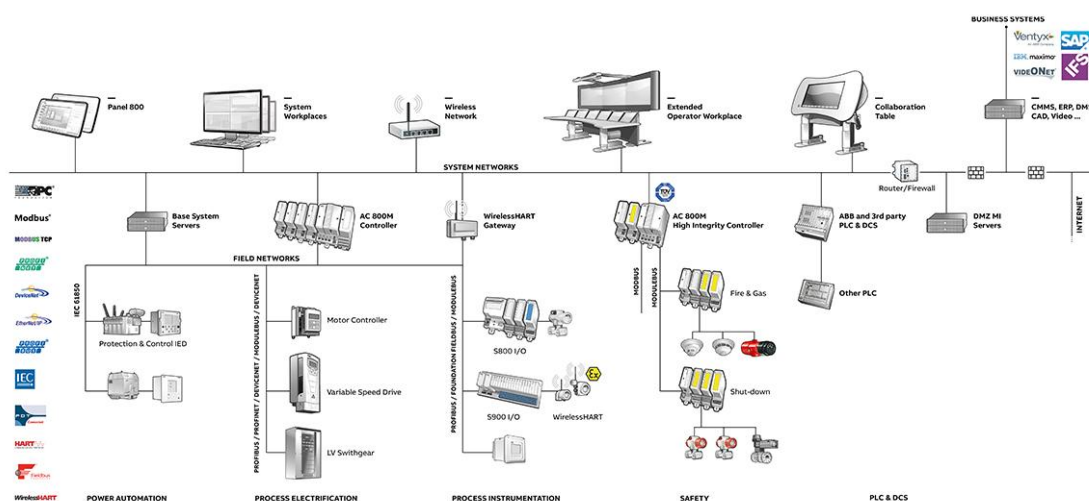
Obrázek 2.11 Část programu pro výstupní hodnoty motoru a senzory [24]

3 Popis nového modelového prostředí

Původní stav hardwarového a softwarového vybavení laboratoře posloužil jako výchozí bod k realizaci komplexnější demo úlohy simulující složitější výrobní celek s využitím obou dostupných PLC a přidáním dotykového panelu. Cílem demo úlohy je propojení hardwarových prostředků pomocí distribuovaného systému řízení ABB 800xA. Tato kapitola popisuje základní vlastnosti a parametry distribuovaného systému a doplňuje informace o dotykovém panelu.

3.1 Distribuovaný systém řízení ABB 800xA

ABB 800xA není jen distribuovaným systémem řízení, ale ve své šíři postihuje také řízení elektrických zařízení a bezpečnostních obvodů. Výhodou 800xA je integrace všech potřebných provozních systémů na jedné platformě nabízející velkou rozšiřitelnost. Orientace na high-profit průmyslová odvětví není u DCS náhodná, systém je schopen monitorovat a řídit miliony tagů s velkou vahou kladenou na bezpečnost a redundanci. Mezi základní výhody patří možnost využití širokého spektra průmyslových komunikačních protokolů, stejně jako možnost zapracování hardwaru jiných výrobců. ABB je kromě jiného schopno nabídnout high-end řešení v oblasti operátorských stanic a kolaboračních panelů. DCS podporuje objektově orientované programování propojené s grafickým editorem pro vytváření pracovních obrazovek s širokou škálou funkcí a rozsáhlou knihovnou šablon a objektů. Samozřejmostí je také pokročilá archivace dat a možnost využití tenkého klienta či webového prostředí.[4]



Obrázek 3.1 Architektura systému ABB 800xA[25]

3.2 Síťová architektura, virtuální počítače a systém 800xA

Na základě požadavků integrace distribuovaného systému řízení 800xA byla přizpůsobena síťová architektura laboratoře požadavkům splňujícím průmyslové standardy simulující skutečné nasazení v provozu. Komunikace mezi síťovými prvky, počítači, dotykovým panelem a PLC probíhá po standardním rozhraní Ethernet. Mezi PLC_1 a frekvenčním měničem je využívána komunikace přes rozhraní ProfiBus.

Z důvodu značných požadavků distribuovaného systému na hardwarovou konfiguraci bylo přistoupeno k využití technologie virtualizace. Na dvou fyzických počítačích lokalizovaných v učebně F204 běží v hypervizech HyperV celkem šest virtuálních strojů simulujících instalaci DCS ve skutečném průmyslovém provozu. Tento stav vystihuje reálné nasazení systému, kdy jsou jeho jednotlivé části rozděleny mezi více serverů z důvodu nutnosti redundance a odolnosti proti výpadkům řízených technologií. Díky virtualizaci je také zaručena delší životnost demo prostředí, které nyní není pevně vázáno na daný lokální počítač, ale může být snadno přeneseno, nakonfigurováno a spuštěno jinde. Tímto krokem by do budoucna nemělo dojít ke ztrátě dat a úloh při nutnosti aktualizace operačního systému nebo výměně počítače.

Na počítači PC1 s Hypervisorem 102 běží Aspect Server ASCS011 (hlavní systémový server – uchovává většinu provozních dat), Engineering Station ES072 (slouží k vytváření kontrolních struktur a programů pro PLC, tvorbě vizualizačních obrazovek atd.) a Operation station OS081 (Operátorská stanice určená pracovníkům provozu). Na počítači PC 2 s Hypervisorem 105 běží History server HIST031 (uchovává vybraná data určená k archivaci nebo dalšímu statistickému zpracování), Engineering station ES071 a Batch server BATCH032 (pro řízení kontinuálních výrobních procesů). Síťové propojení jednotlivých virtuálních počítačů a další detaily zapojení, stejně jako přístup k připojeným PLC a kontrolnímu panelu je zřejmý z infografiky dodané firmou ABB spolu s instalací systému. Všechny virtuální počítače obsahují několik uživatelských účtů. Účet 800xAInstaller se využívá při instalaci systému a jeho jednotlivých částí, účet 800xAService je určen pro IT podporu a účty DemoA, DemoE, a DemoO. jsou určeny pro inženýrský a operátorský přístup do systému.

Obrázek 3.2 Lokální struktura systému 800xA pro demo úlohy[16]

3.3 Control Panel 800

Pro rozšíření možností prezentace a ovládání demo úloh byl do nového prostředí přidán vizualizační dotykový display PP874 z řady ABB Panel 800. Panel je vybaven TFT/LED barevným displejem s vysokým rozlišením usnadňující interakci se zařízením. Výhodou panelů z řady 800 jsou integrované grafické a programové knihovny. K programování panelu slouží program Panel Builder 800, který je spolu s dalšími rozšiřujícími moduly systému 800xA nainstalován v inženýrské stanici na virtuálním počítači. Tabulka 3.1 Vybrané parametry panelu PP874 uvádí důležité vlastnosti dotykového panelu.[29]

Tabulka 3.1 Vybrané parametry panelu PP874 [29]

Parametr	Hodnota
Číslo panelu	PP874
Velikost displeje	7"
Rozlišení	800x480 (16:9)
Paměť	80 MB
Světlost displeje	350 cd/m2
Hlavní rozměry	204x143x49 mm
Váha	0,8 Kg
Síťové připojení	1x 10/100 Mbps Ethernet
Sériové porty	2x (RS232/RS485/RS422)
Další výbava	Slot na SD karty, USB



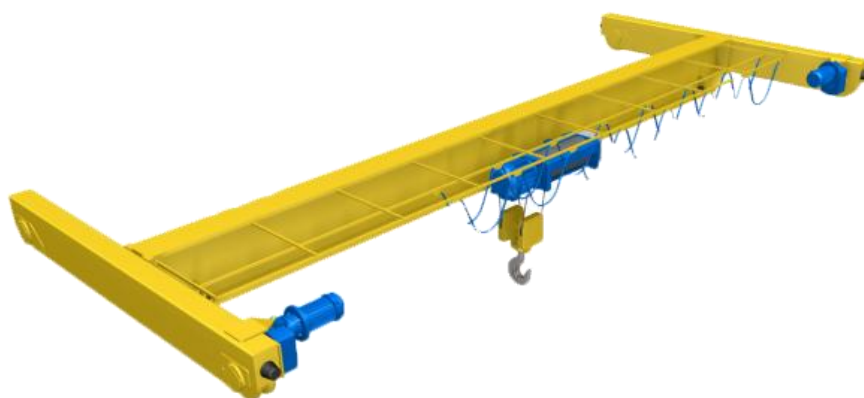
Obrázek 3.3 HMI PP874 řady Panel 800 [29]

4 Nová demo úloha

Nová demo úloha bude využívat asynchronní motor, který je jedním z nejpoužívanějších pohonných ústrojí nejen v průmyslové praxi. Mezi zásadní přednosti asynchronních motorů patří vysoká účinnost, absence opotřebitelných komutátorů a možnost řízení v současnosti rozvinutými frekvenčními měniči. S asynchronními motory se setkáváme u ventilátorů, vzduchotechniky, různých druhů čerpadel, kompresorů, dopravníků, jeřábů, výtahů, soustruhů a dalších výrobních strojů, kde není od pohonu vyžadována absolutní přesnost při polohování.

Použitý motor v současnosti nedisponuje žádnou nastavitelnou zátěží v podobě dynamického měniče, ani jinou použitelnou a bezpečnou technologií. Všechny hodnoty vstupů pro neexistující technologie tak budou při provozu demo úlohy dosaženy simulačně.

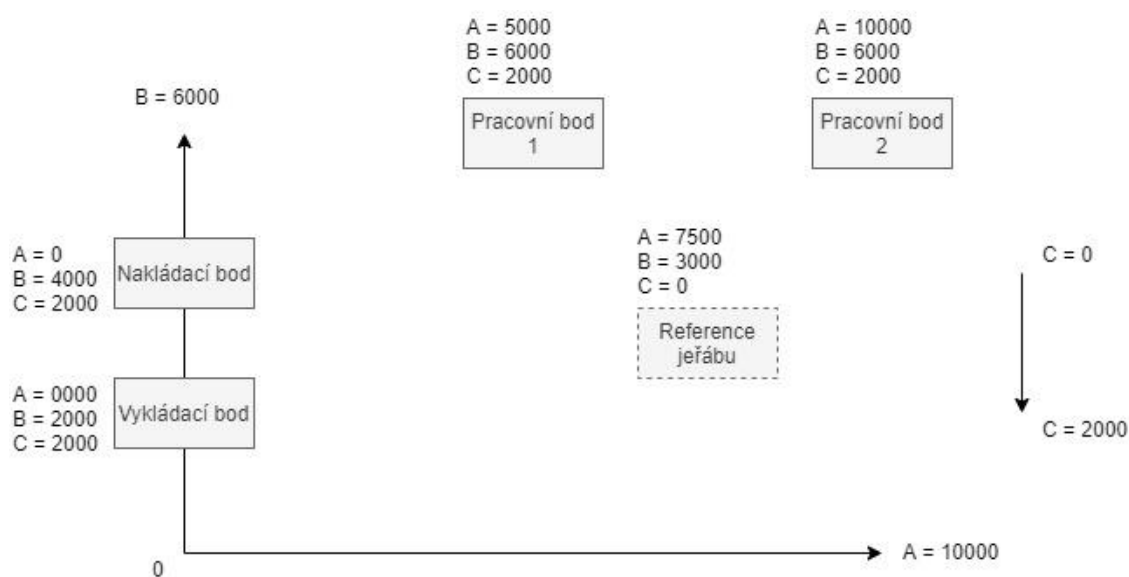
Analogové a digitální vstupy a výstupy jsou v současnosti stále základním stavebním kamenem při řízení strojů a zařízení v průmyslové výrobě. Přestože se v posledních letech stále častěji nasazují smart senzory, například na komunikaci IO-Link, které již ze své podstaty nabízejí široké možnosti v rámci zpracování naměřených nebo řídicích hodnot, obsahují pokročilé nástroje diagnostiky, nebo přispívají k zjednodušení cable-managemantu v rozvaděčích strojů, klasické dumb vstupy a výstupy si stále drží své postavení na trhu a nedá se předpokládat, že by byli v nejbližší době plně nahrazeny smart technikou. Také nová demo úloha bude obsahovat ovládací prvky pracující s analogovými a digitálními signály.



Obrázek 4.1 Příklad mostového jeřábu[30]

Popis demo úlohy

Nová demo úloha představuje výrobní halu vybavenou mostovým jeřábem, který slouží k přesunu výrobků mezi důležitými body v prostoru. Jeřáb přijede do nakládacího bodu, naloží výrobek a následně jej zaveze k jednomu z volných strojů v pracovním bodě, kde je výrobek dále zpracován. Po dokončení výrobní operace opět přijíždí jeřáb a odváží výrobek do vykládacího bodu, odkud míří k expedici nebo dalšímu zpracování. Jednotlivé osy jeřábu jsou poháněny asynchronním elektromotorem. Jelikož je pro úlohu fyzicky dostupný pouze jeden motor, tak budou pohyby všech tří motorů simulovány na jednom fyzickém motoru pomocí změny referenčních otáček a referenčního momentu.



Obrázek 4.2 Schéma modelové výrobní haly s významnými body

Funkční specifikace demo úlohy

Úloha bude obsahovat ovládání jeřábu ve třech osách. Jeřáb bude možné ovládat ručně z operátorské stanice nebo pomocí tlačítek a potenciometrů na rozvaděči. V ručním režimu bude možné vybrat osu k pohybu, zadat požadovanou polohu a spustit či zastavit pohyb jeřábu. Po najetí do příslušného bodu v ručním režimu bude možné naložit nebo vyložit výrobek. Jeřáb bude možné uvést do automatického módu. Automatický mód se bude starat o automatický rozvoz výrobků mezi jednotlivými pracovními body ve výrobní hale na základě dané báze pravidel. Mezi automatickým a manuálním módem bude možné libovolně přepínat. Přepínání mezi módy bude možné za provozu a přechod neovlivní stav výrobků a dalších běžících procesů. Po přepnutí z manuálního do automatického módu najede jeřáb nejprve bezpečným způsobem do referenční pozice. Jeřáb se z důvodu simulace více pohonů bude pohybovat vždy jen v jedné ose, pohyb ve více osách současně nebude možný. Zadávaní výrobku bude probíhat v přípravném bodu. Název výrobků a stanovený výrobní čas bude možné zadat ručně nebo automaticky

vygenerovat. Při ručním zadávání bude hlídán správný tvar zápisu parametrů. Z přípravného bodu bude výrobek pokračovat do fronty ústící v nakládacím bodu jeřábu. Z přípravného bodu, bodů fronty a nakládacího bodu bude možné výrobek smazat. Výrobky ve výrobních bodech nelze smazat, ale bude možné přerušit výrobu a označit výrobky jako nesprávné. Po vyložení výrobku ve vykládacím bodu bude navýšeno počítadlo správných, případně nesprávných kusů. Počítadlo bude možné vynulovat. Všechny operace jeřábu bude možné zastavit tlačítkem Central Stop. Všechny výrobní body bude možné resetovat do výchozího nastavení. Reálný motor bude možné při práci programu vypnout, jeho provoz nebude pro provoz programu nutný.

Použité hardwarové komponenty

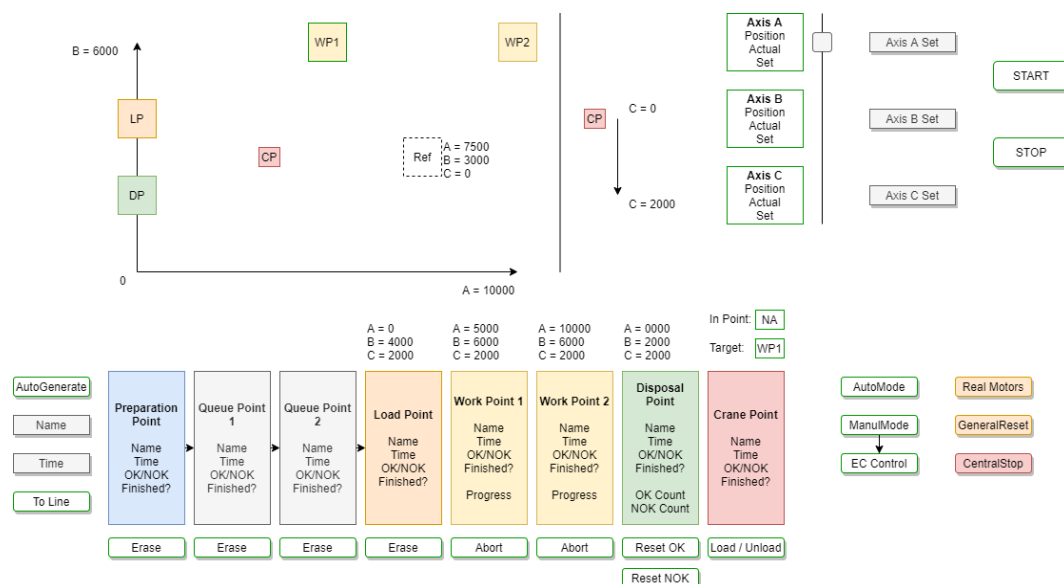
Pro realizaci demo úlohy budou využity obě dostupná PLC ABB AC800M. Hlavní program poběží na PLC_1, ke kterému bude zároveň přes PROFIBUS připojen frekvenční měnič ABB ACS800 s asynchronním motorem. Na PLC_2 poběží program pro zpracování hodnot ze tří tlačítek a tří potenciometrů na rozvaděči. Využity budou také signální lampy umístěné v tlačítkách. Dále bude pro zobrazování důležitých údajů využit HMI Panel 800. Využity budou oba dostupné počítače, na kterých poběží systém 800xA.

Fyzikální veličiny, datové typy a inženýrské jednotky

Program bude obsahovat proměnné držící hodnoty parametrů motorů, pozice os, stavu výrobků a pracovišť. Podrobný výčet použitých proměnných, jejich datových typů a popis významu je uveden v další části práce.

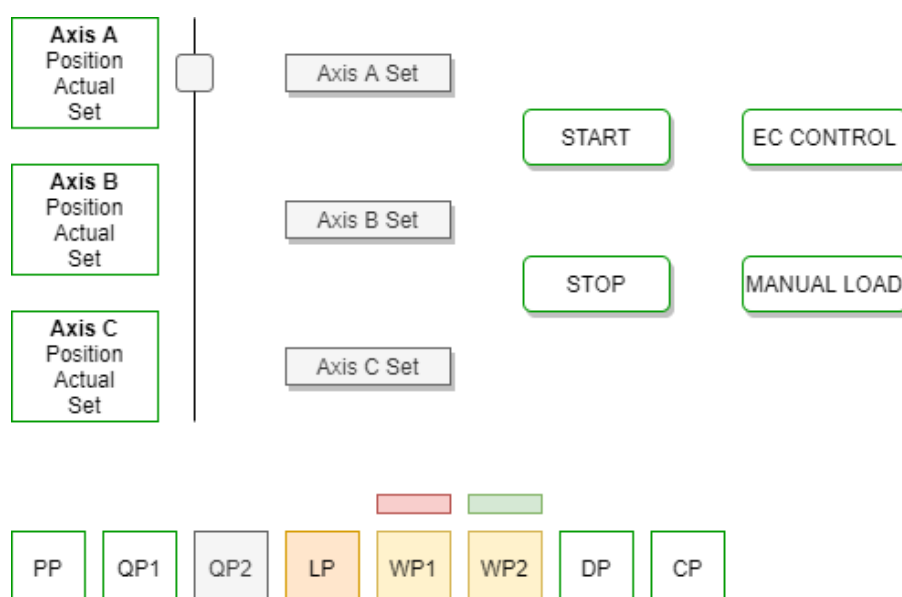
HMI demo úlohy

Úlohu bude možné řídit pomocí operátorských obrazovek na stolním počítači, pomocí obrazovek Panelu 800 a pomocí světelné signalizace na rozvaděči. V případě kontroly z operátorské stanice bude mít operátor k dispozici všechny informace o stavu výrobků v jednotlivých bodech, na displeji uvidí aktuální pozici jeřábu a bude mít k dispozici všechna funkční tlačítka a zadávací pole.



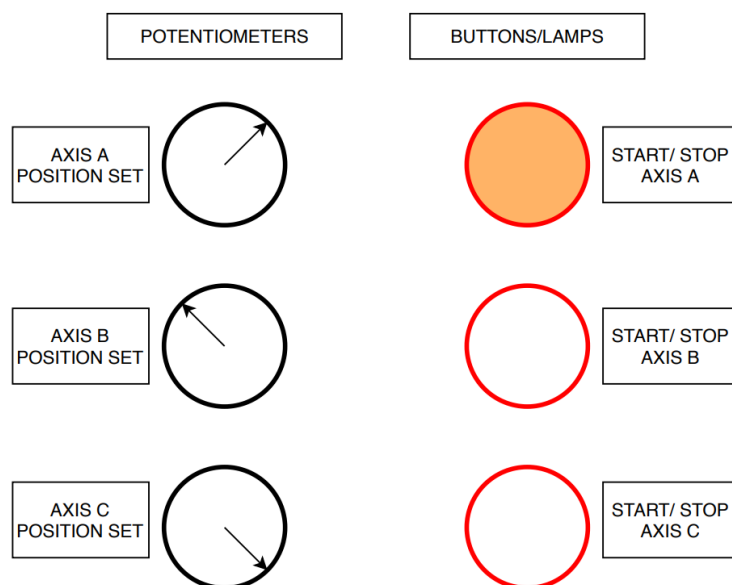
Obrázek 4.3 Návrh obrazovky operátorské stanice

Při zapnutí manuálního ovládání bude mít uživatel možnost ovládat jeřáb z HMI Panelu s omezenou nabídkou tlačítek a zadávacích polí. Uvidí také zjednodušený stavový indikátor jednotlivých bodů.



Obrázek 4.4 Návrh displeje HMI panelu

V okamžiku, kdy bude zapnuto ovládání skrze tlačítka a potenciometry na rozvaděči, zůstane na panelu v provozu pouze tlačítko Manual Load sloužící k naložení a vyložení výrobku. Zbytek tlačítek nebude možné stisknout, protože nastavování pozice a spouštění jeřábu přejde na rozvaděč. Jednotlivými potenciometry bude nastavována požadovaná poloha. Při stisknutí tlačítka se spustí posun v dané ose. Pokud osa dojede až do zadané polohy, tlačítko se rozsvítí. Pokud bude tlačítko stisknuto znova ještě před dojetím do zadané polohy, posun v ose se zastaví.



Obrázek 4.5 Návrh využití potenciometrů a tlačítek na rozvaděči

Referenční úlohy a obdobné realizace

Účelem demo úlohy je představení distribuovaného systému řízení a ukázka možností propojení jednotlivých zařízení, PLC, HMI panelů, snímačů a výkonových komponent. Úloha vychází z reálných aplikací, avšak zavádí určité zjednodušující předpoklady a postupy. Pokud by byl program dále modifikován, mohl by být využit jak pro řízení reálného jeřábu, tak například tříosého manipulátoru. Níže uvedené referenční práce mohou být využity pro prohloubení znalostí o způsobu řízení a polohování podobných zařízení: *Model automatického parkovacího domu a jeho řízení* [31], *Řízení pojezdu mostového jeřábu pomocí frekvenčního měniče prostřednictvím programovatelného automatu Simatic S300* [32], nebo *Model mostového jeřábu*. [33]

Názvosloví a terminologie

V celé diplomové práci a demo úloze je odkazováno na tvorbu aplikace pro jeřáb, avšak vypracovaná demo úloha může být stejně tak použita pro tříosí manipulátor, nebo jiný podobný mechanismus. Demo úloha se nesoustřeďuje na řešení řízení motoru v závislosti na skutečných fyzikálních podmínkách. Všechny číselné hodnoty jednotek vzdáleností jsou uvedeny v milimetrech.

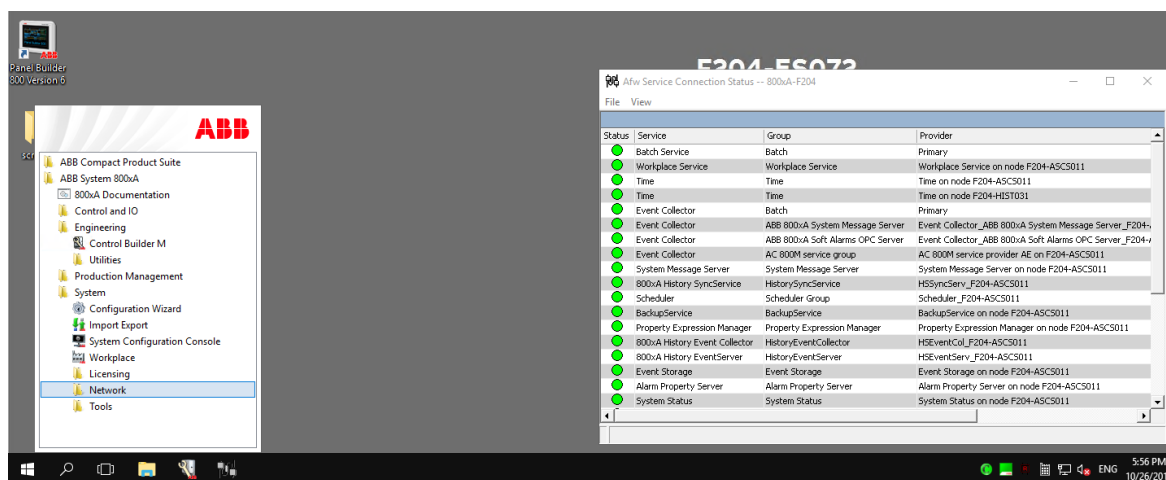
Volba způsobu ovládání motoru

Na základě dostupného hardwaru a celkovému směřování demo úlohy bylo rozhodnuto o ovládání motoru pomocí výpočtu teoretické doby jeho běhu potřebné pro přesun z výchozího do požadovaného bodu v dané ose. Motor je na vypočtenou dobu zapnut a po uplynutí času zase vypnut a do aktuální polohy osy zapsána nová teoretická hodnota. Tento způsob ovládání motorů je z pohledu demo aplikace dostatečný, ovšem pokud by aplikace měla být využita k řízení reálného tříosého systému, musela by být část programu modifikována tak, aby na základě údajů ze snímače otáček řídila požadovanou polohu například pomocí softwarového PID regulátoru obdobně, jako u servo motorů. Pro tento způsob řízení by však bylo nutné měnit zapojení snímače otáček ve frekvenčním měniči, kdy by vývody ze snímače byly připojeny do speciálního modulu pro snímání pulzů připojenému přímo k PLC. Tato úprava by však vedla k znefunkčnění původního Control Modulu, který by nemohl být nadále využíván v případných dalších aplikacích a demo úlohách. Na základě této skutečnosti a zjištění, že zmíněná pulzní karta není funkční, bylo rozhodnuto realizovat ovládání motoru skrze časové proměnné, kdy skutečný stav softwarově simulovaných os nebude závislý na přesném počtu otáček hřídele motoru. V případě reálné aplikace by bylo kromě výše uvedených změn v softwaru a hardwaru potřeba doplnit řízení elektrohydraulických brzd jednotlivých pojezdů jeřábu, zahrnout referenční a koncová čidla a vyřešit výpočty a kontrolu dynamických vlastností systému v závislosti na změně zátěže jeřábu podobně, jak je uvedeno ve výše zmíněných referenčních pracích.

5 Realizace demo úlohy – Příprava a nastavení Hardwaru

Realizace demo úlohy v distribuovaném systému řízení ABB 800xA se dá rozdělit do dvou význačných etap. Do první etapy spadá konfigurace hardwaru a tvorba řídicí logiky a do druhé etapy spadá vizualizace a konfigurace dalších součástí DCS. Téměř veškerá inženýrská činnost bude vykonávána na virtuálním počítači Engineering Station (ES 072) pod uživatelským účtem DemoA.

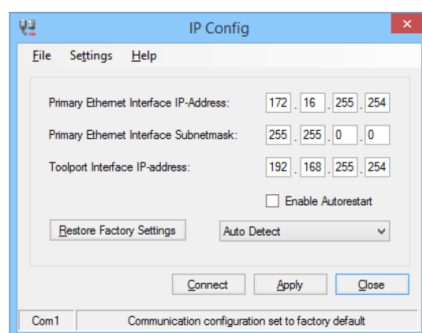
Základním rozcestníkem systému 800xA je ABB Start Menu, odkud je možné přistoupit k jednotlivým aplikacím (na obrázku vlevo). Příslušná ikona se nachází na ploše virtuálního počítače. Další důležitou součástí je monitor stavu systému nacházející se v pravém dolním rohu obrazovky. Po rozkliknutí zelené ikony je možné podrobněji prohlížet stav jednotlivých částí systému. Pro správnou funkci systému musejí být všechny stavové indikátory zelené. Pokud je vše v pořádku, je možné v systému začít pracovat. V případě, že je u některých součástí systému hlášeno varování, nebo dokonce chyba, je této části nutné věnovat pozornost a před započatím prací zjednat nápravu. Následující popis přípravy hardwaru a Control projektu platí jak pro PLC_1, na kterém je proces dále popisován, tak pro PLC_2, u kterého byl proces přípravy totožný.



Obrázek 5.1 ABB Start Menu a 800xA System Status Overview

Nastavení IP adres PLC

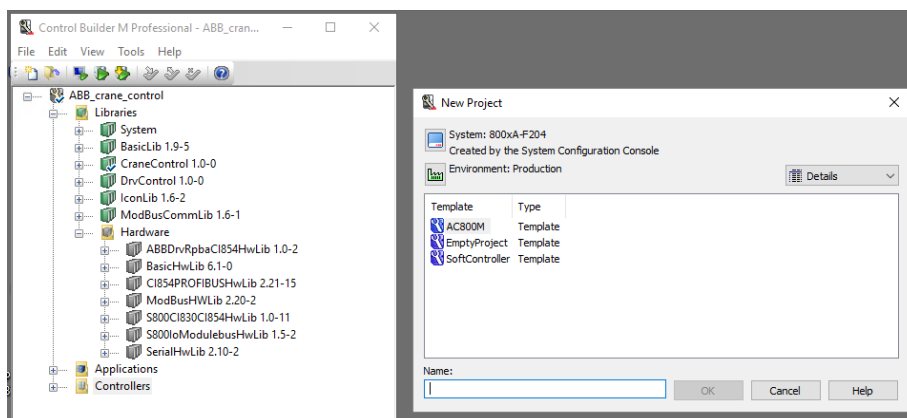
Jako první musí být nastaveny IP adresy jednotlivých PLC. K tomuto účelu slouží nástroj IPConfig přístupný z ABB StartMenu. Po otevření konzoly je potřeba připojit PLC přes konfigurační COM4 napřímo k PC, ze kterého je prováděna konfigurace. Po fyzickém propojení se stiskne tlačítko *Connect* a následně je potřeba podržet Inicializační tlačítko na PLC, dokud se nerozblíká dioda Run. Následně dojde k připojení a do konzoly se vypíše defaultní IP adresa. Adresu můžeme nyní nastavit podle vlastní potřeby, aplikovat a konzolu zavřít. Pro PLC 1 byla nastavena IP adresa 172.16.80.152 a pro PLC 2 adresa 172.16.80.162. [27],[34]



Obrázek 5.2 Nastavení adresy pomocí IP Config

Vytvoření nového projektu a připojení knihoven

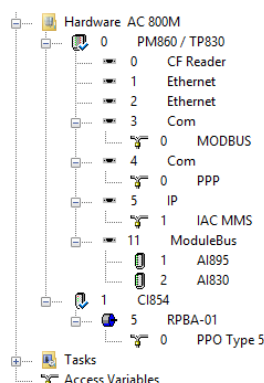
Po nastavení IP adres spustíme přes ABB StartMenu program Control Builder M sloužící k programování PLC a vytváření logiky systémového řízení. Po spuštění programu byl vytvořen nový projekt pro AC800M pojmenován ABB_crane_control. Ze všeho nejdříve je nutné do projektu vložit knihovny. Jejich výčet je viditelný na obrázku níže. V zásadě se jedná o systémové knihovny a knihovny s ovladači pro jednotlivé moduly připojené k PLC, případně pro komunikaci. Za povšimnutí stojí také ikony pro testování, online režim a nahrání projektu do PLC v podobě ikon s variacemi značky play v horní části programu.



Obrázek 5.3 Okno pro vytvoření nového projektu a výčet použitých knihoven

Konfigurace PLC v programu Control Builder

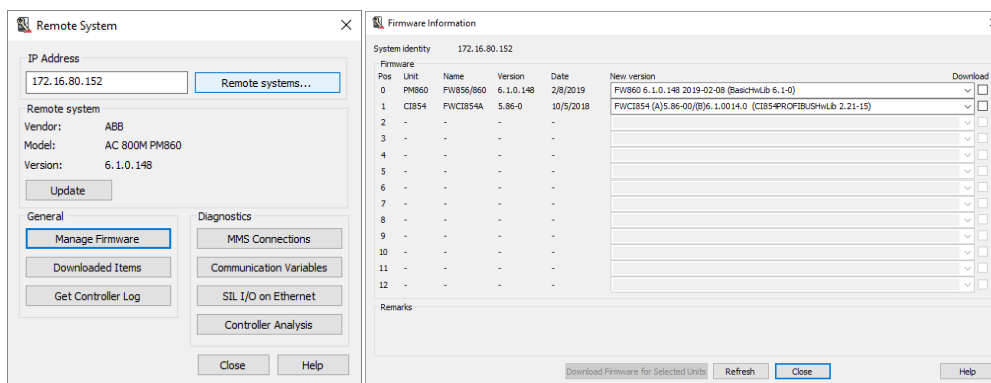
Po úspěšném přidání knihoven bylo potřeba nakonfigurovat kontrolér dle skutečného stavu. Kontrolér byl pojmenován PLC_1, byl vybrán příslušný typ dle procesorové jednotky a byla mu nastavena IP adresa. Dále byly k PLC připojeny knihovny pro jednotlivé moduly. Jako poslední byly přidány položky hardwaru do struktury PLC stejným způsobem, jako jsou zapojeny fyzicky. Pro připojení modulů stačí kliknout na příslušnou větev, vybrat možnost *Insert Unit* a následně v nabídce vyhledat konkrétní modul.



Obrázek 5.4 Konfigurace hardwaru pro PLC 1

Aktualizace firmwaru PLC

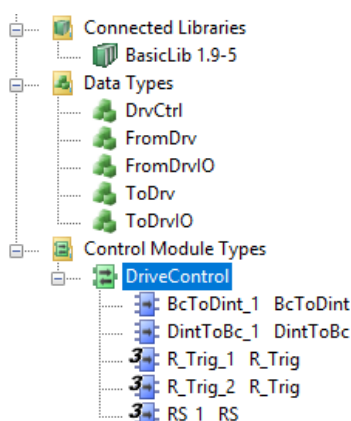
Po vytvoření hardwarové konfigurace bylo potřeba aktualizovat firmware obou PLC a jejich součástí. Tato možnost je dostupná z nabídky Tools -> Maintenance -> Remote System. V okně Remote system je potřeba zadat příslušnou IP adresu a stisknou tlačítko Update. Pokud se načtou informace o systému, je možné s ním dále pracovat a například spravovat firmware, procházet nahrané soubory či provádět diagnostické úkony. Po kliknutí na položku *Manage Firmware* se otevře okno s přehledem o současném firmwaru a pokud existuje jeho novější verze, nabídne ji k aktualizaci. V okně byl vybrán firmware pro PLC a frekvenční měnič a kliknutím na tlačítko *Download to Selected Unit* byl nový firmware nahrán do PLC a frekvenčního měniče. Dokončením tohoto kroku bylo PLC připraveno k dalšímu používání.



Obrázek 5.5 Aktualizace firmwaru

Přenos původní aplikace do nového prostředí

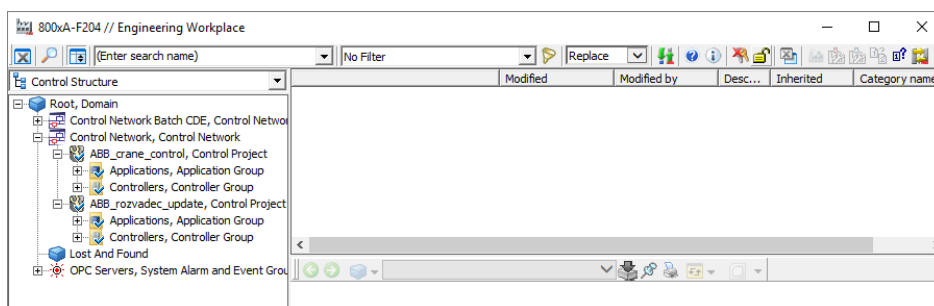
Původní knihovna s aplikací pro řízení motoru vytvořená v Compact Control Builderu, která byla dostupná z PC 1 musela být na základě nemožnosti importu do systému 800xA vytvořena znova. Přímý import nebylo možné realizovat z důvodu licenčních politiky ABB. Z toho důvodu byla knihovna DrvControl přepsána jedna ku jedné a uložena v knihovní struktuře 800xA. Nyní je dostupná pro všechny uživatele a v případě potřeby může být exportována/importována i do jiné instance systému. Control modul byl shodně naprogramován v jazyce Function Block Diagram, stejně jako byly přeneseny i datové typy tvořící kontrolní a stavové slovo určené pro komunikaci s frekvenčním měničem.



Obrázek 5.6 Struktura uživatelské knihovny DrvControl

Propojení PLC pomocí systému 800xA

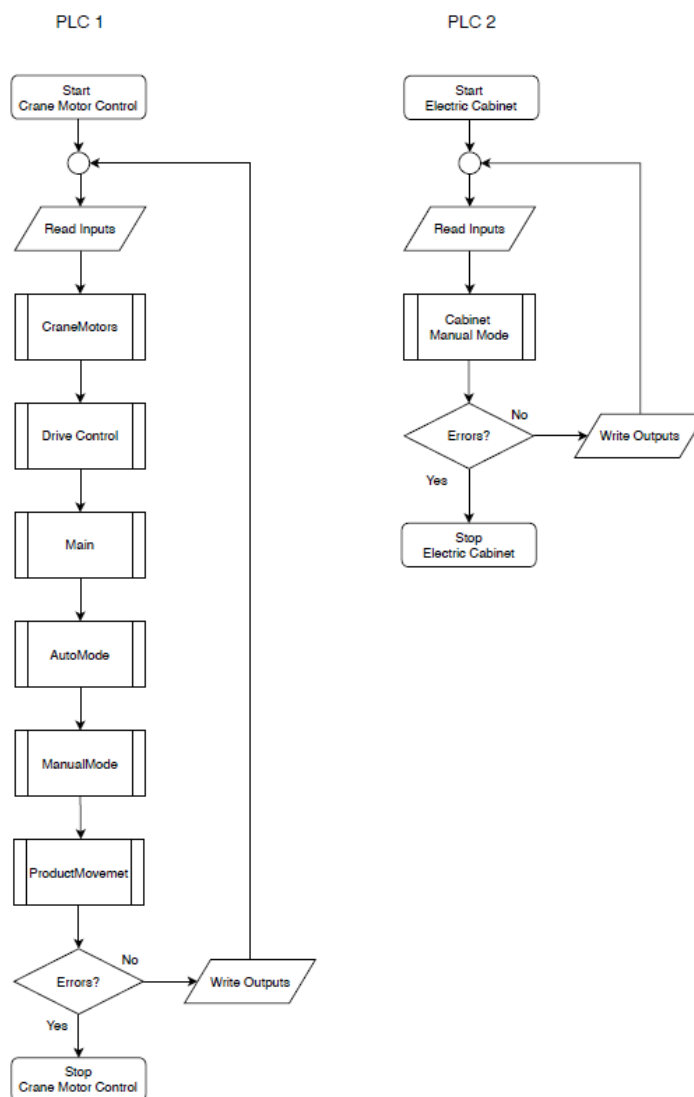
Rozdíl mezi prací v Compact Control Builderu a jeho plné verzi v rámci DCS systému je v tom, že v plné verzi nejsou vytvářeny projekty samostatně, ale jsou součástí vyššího systému. Ke správě 800xA slouží nástroj Engineering Workplace dostupný z ABB StartMenu, po jeho spuštění máme k dispozici nástroj pro správu jednotlivých struktur systému, o kterých bude pojednáváno dále. Pro dosavadní postup práce bylo důležité, že po vytvoření projektů a konfiguraci obou PLC jsou tyto projekty viditelné v kontrolní struktuře systému na stejné síti, což je předpokladem pro jejich vzájemnou komunikaci a výstavbu dalších součástí systému včetně vizualizační obrazovky. [34]



Obrázek 5.7 Kontrolní struktura v prostředí Workplace

6 Realizace demo úlohy – Proměnné a vývojový diagram

Na základě zapojení dostupné hardware bylo rozhodnuto, že hlavní program pro ovládání jeřábu v automatické i manuální sekvenci bude lokalizován na PLC_1 s frekvenčním měničem a motorem, kdežto PLC_2 bude obsahovat podpůrný program pro vstupy ovládání z tlačítek a potenciometrů v rozvaděči v ručním režimu. Kompletní vývojový diagram včetně seznamu proměnných je dostupný v příloze diplomové práce. Níže jsou rozebrány klíčové části jednotlivých programů a vysvětlen princip fungování důležitých smyček. Kontrolní modul Drive Control byl vygenerován z převzaté knihovny DrvControl přenesené do systému 800xA z demo úlohy pro Compact Control Builder popsané v předchozí kapitole. Princip ovládání demo úlohy je možné blíže nastudovat v uživatelské příručce – příloha D.



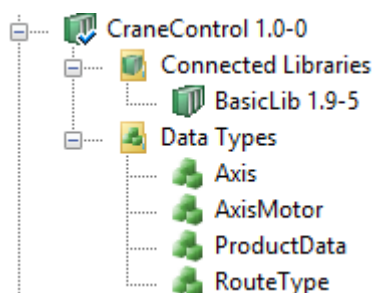
Obrázek 6.1 Přehled programů v jednotlivých PLC

6.1 PLC 1 – Aplikace Crane Motor

Hlavní aplikace demo úlohy obsahuje důležité globální proměnné uchovávající informace o stavu výrobků v jednotlivých bodech. Další proměnné slouží k výběru mezi jednotlivými módy aplikace a zajišťují komunikaci s frekvenčním měničem. Pro potřeby uchování informací o stavu jednotlivých bodů byl vytvořen nový datový typ s názvem ProductData obsahující název výrobku, čas výroby, status správnosti výroby a indikaci dokončení výroby. Informace o výrobku vstupujícím do aplikace skrze Přípravný bod se poté v daný okamžik přelévají mezi vybranými body. Podobným způsobem je definován také datový typ Axis držící současné, nastavené a další pracovní hodnoty jednotlivých os. Datový typ Axis Motor pak obsahuje informace k simulovaným motorům a datový typ RouteType zase strukturu jednotlivých kroků pro automatický režim.

Tabulka 6.1 Globální proměnné v aplikaci Crane Control v PLC 1

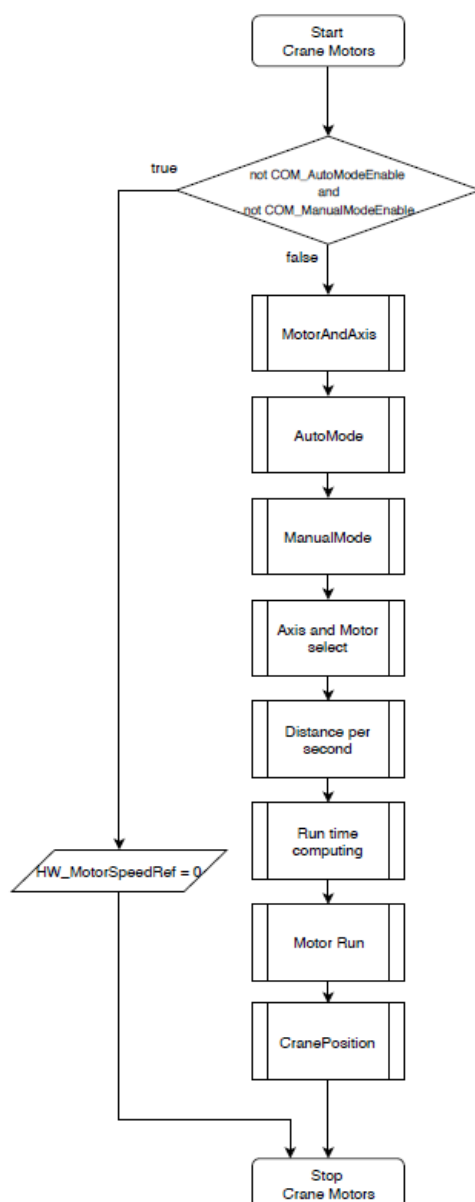
Název	Datový typ	Ini. hodnota	Jednotka	Popis
gvMotorComm	DrvCtrl		[-]	Komunikace s frekvenčním měničem
gvPP	ProductData		[-]	Preparation Point
gvQP1	ProductData		[-]	Queue Point 1
gvQP2	ProductData		[-]	Queue Point 2
gvLP	ProductData		[-]	Loadind Point
gvDP	ProductData		[-]	Disposal Point
gvCP	ProductData		[-]	Crane Point
gvWP1	ProductData		[-]	WorkPoint 1
gvWP2	ProductData		[-]	WorkPoint 2
gvResetP	ProductData		[-]	Reset Point
gvAutoModeEnable	bool	false	[-]	Zapne AutoMode
gvCraneInPosition	bool		[-]	Indikuje jeřáb ve známé pozici
gvManualModeLoad	bool	false	[-]	Manuální naložení/vyložení produktu
gvAutoGenerateProduct	bool		[-]	Zapne automatické generování produktu
gvRealMotors	bool	false	[-]	Zapne reálný motor
gvManualModeEnable	bool	false	[-]	Zapne ManualMode



Obrázek 6.2 Uživatelská knihovna datových typů CraneControl

Funkční Diagram Crane Motors

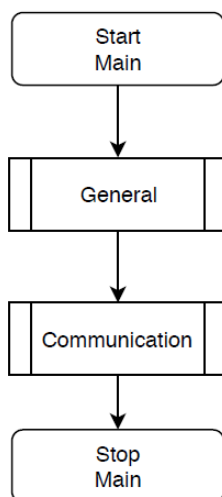
Funkční diagram Crane Motors slouží k pohybu jeřábu v jednotlivých osách a je aktivní, pokud je v aplikaci spuštěn manuální nebo automatický režim. Podprogram MotorAndAxis slouží k základnímu nastavení parametrů os a motorů a zajišťuje komunikaci s PLC_2 v rozvaděči. Podprogramy AutoMode, případně ManualMode se starají o zadání požadované pozice ve vybrané ose a následující část kódu se postará o výpočet potřebného posunutí. Podprogram MotorRun ovládá výstupní otáčky motoru a jako poslední je vyhodnocována informace o tom, zda se jeřáb nachází v jedné z výše definovaných pracovních pozic.



Obrázek 6.3 Vývojový diagram kontrolního diagramu Crane Motors

Program Main

Program Main slouží k zdefinování důležitých funkčních podmínek při práci s globálními proměnnými a k nastavení komunikace mezi vstupními a výstupními moduly, frekvenčním měničem a komunikačními proměnnými sloužícími k předávání hodnot mezi různými PLC na jednom kontrolním networku a také pro předání hodnot do funkčních diagramů. Podprogram General řeší mimo jiné funkci Central Stopu, znemožňuje současné zapnutí automatického a manuálního módu a také uvede do provozu skutečný motor, nebo jej naopak vypíná.

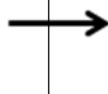


Obrázek 6.4 Program Main

Program AutoMode

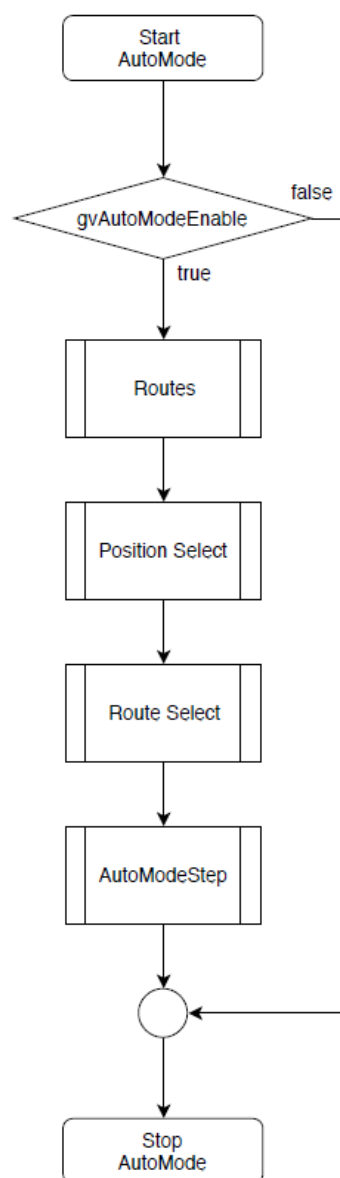
Program má za úkol řídit funkci automatického režimu a při své práci komunikuje s příslušným podprogramem v diagramu Crane Motors. Pokud je povolen automatický mód, tak program ve svém prvním běhu v podprogramu Routes nadefinuje pole cest dle uživatelského datového typu RouteTypes, kdy každá cesta obsahuje celkem pět definovaných pohybů potřebných k přesunu jeřábu mezi dvěma místy.

Program AutoMode		Podprogram AutoMode v Control Diagramu					
RoutesArray	Pohyb	Pohyb	Osa X	Pohyb	Osa Y	Pohyb	Osa Z
Route1	Route1[1]=30	10	0	20	0	30	0
...	Route1[2]=22	11	5000	21	2000	31	2000
RouteX	Route1[3]=10	12	7500	22	3000		
	Route1[4]=23	13	10000	23	4000		
	Route1[5]=31			24	6000		



Obrázek 6.5 Výběr osy a nastavení cílové pozice skrze číslo pohybu

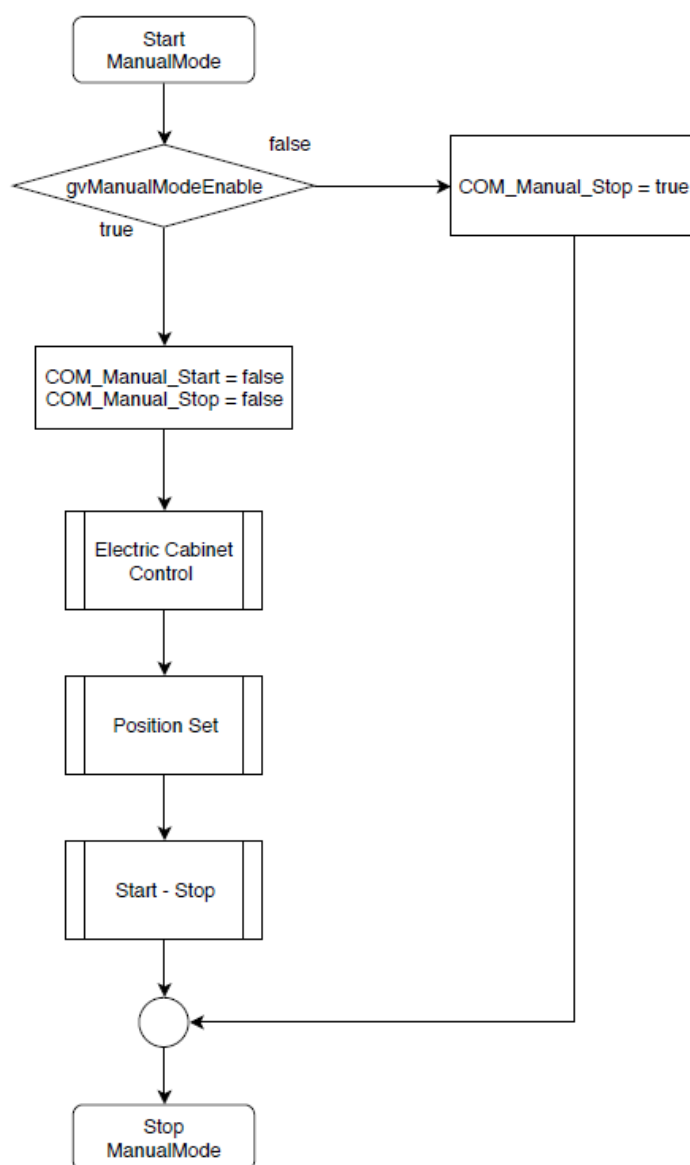
Podprogram Position Select je určen k definování dalšího cíle jeřábu na základě současného stavu jednotlivých pracovních bodů ve výrobní hale. Všechny možné a zakázané varianty stavů jednotlivých bodů a výsledný výběr cesty je uveden v pravdivostní tabulce v příloze DP. Následující část kódu pak vyhodnocuje, zda jeřáb již dorazil do vybrané pozice a předal výrobek a pokud se tak stalo, dá pokyn k výběru nové cesty jeřábu. Poslední část programu slouží ke komunikaci s podprogramem v Control Diagramu, kterému předá informaci o požadovaném pohybu a po potvrzení o vykonání pohybu přejde na další krok ve zvolené cestě. Jakmile dokončí celou cestu, je možný výběr nové cesty a celý proces se opakuje.



Obrázek 6.6 Vývojový diagram programu AutoMode

Program ManualMode

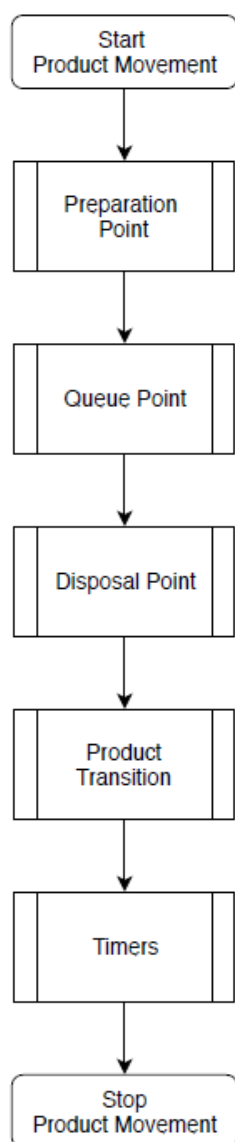
V případě volby ovládání jeřábu v manuálním módu je aktivován příslušný program zajišťující možnost ručního výběru osy, zadání požadované pozice a spuštění či zastavení pohybu. Kromě ručního zadání parametrů na operátorské obrazovce je zde při aktivaci rozvaděčového módu řešen také přenos příslušných dat z PLC 2. Podprogram Position Set kromě vyčítání příslušných hodnot řeší také nastavení limitů zadání, aby se jeřáb nedostal mimo svůj operační rádius. Poslední podprogram řeší vypínání a zapínání pohybu ve vybrané ose.



Obrázek 6.7 Vývojový diagram programu ManualMode

Program ProductMovement

Pro správu obsahu jednotlivých bodů a přesuny mezi nimi slouží program ProductMovement. V podprogramu Preparation Point je realizováno automatické generování produktů a zároveň také kontrola ručního zadávání, před posunem výrobku do řady k nakládacímu bodu. Další podprogramy umožňují mazání a resetování jednotlivých bodů, počítají správně a nesprávně vyrobené kusy a starají se o spuštění a zastavení časovačů ve výrobních bodech. Důležitý je pak podprogram Product Transition obsahující podmínky pro naložení a vyložení výrobků v ručním a automatickém režimu.



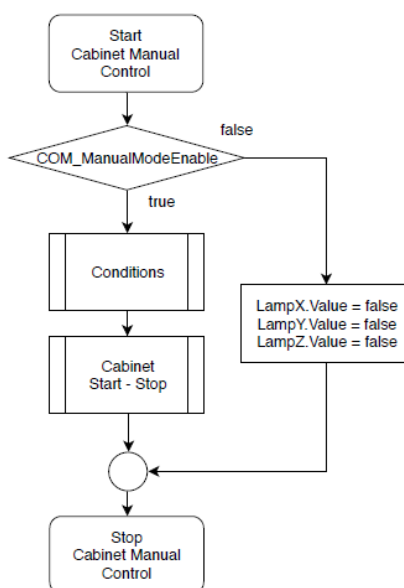
Obrázek 6.8 Vývojový diagram programu ProductMovement

6.2 PLC 2 – Aplikace Rozvaděč

Druhé PLC plní v demo úloze spíše podpůrnou roli a jeho programová stránka je podstatně jednodušší než u prvního PLC. Zatímco PLC_1 zpracovává všechny důležité programy týkající se pohybu motorů a udržuje informace o stavu jednotlivých pracovních bodů a os, PLC 2 slouží pro vyčítání analogových a digitálních vstupů a signalizaci pomocí lamp.

Program Cabinet Manual Mode

Program Cabinet Manual Mode je určen pro ovládání jeřábu pomocí tlačítek a potenciometrů. Pokud je aktivován manuální mód v PLC 1 a v programu ManulMode v PLC 1 byla aktivována kontrola skrze rozvaděč, jsou v podprogramu Conditions požadované pozice jeřábu v jednotlivých osách vypočítávány z aktuálních hodnot analogových vstupů. Dle pootočení potenciometrů teče do vstupní analogové karty proud v rozmezí od 0 do 24mA. Příchozí hodnota je přepočtena na procentuální podíl z rozsahu a dále zpracována programem, který procentuální hodnotu vynásobí s výpočetními konstantami tak, aby maximální hodnota potenciometru udávala maximální možnou hodnotu pozice příslušné osy. Konstanty jsou z důvodu rozlišitelnosti a citlivosti potenciometrů nastaveny tak, aby požadovanou hodnotu měnili v řádu stovek milimetrů. Po nastavení příslušné hodnoty potenciometrem podprogram Cabinet Start-Stop čeká na startovací pokyn při stisknutí tlačítka. Pokud je tlačítko stisknuto znovu, rozjetý pohyb se zastaví v aktuální pozici a je možné spustit pohyb v jiné ose, nebo dokončit započatý pohyb. Pokud jeřáb najede do nastavené pozice, rozsvítí se lampa v příslušném ovládacím tlačítku.



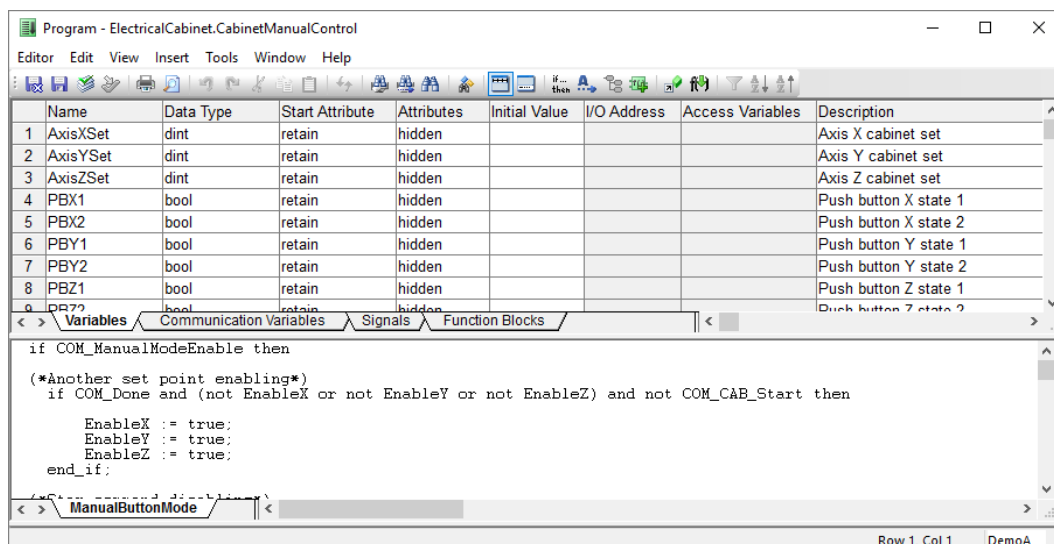
Obrázek 6.9 Vývojový diagram programu CabineManualControl

7 Realizace demo úlohy – Programování v Control Builderu

Na základě vývojových diagramů popsaných v předchozí kapitole bylo přistoupeno k realizaci programového řešení demo úlohy v softwaru Control Builder. Základní konfigurace projektů do stavu připravenosti pro samotné programování kontrolní logiky byla popsána v páté kapitole. Následující kapitola shrnuje základní poznatky k práci s Control Builderem při vytváření aplikace. Control Builder M integrovaný v DCS 800xA disponuje možností programování ve standardních programovacích jazycích používaných u PLC. Pro pohodlnější týmovou práci většího množství programátorů na jednom projektu disponuje program možností rezervace vybrané části projektu.

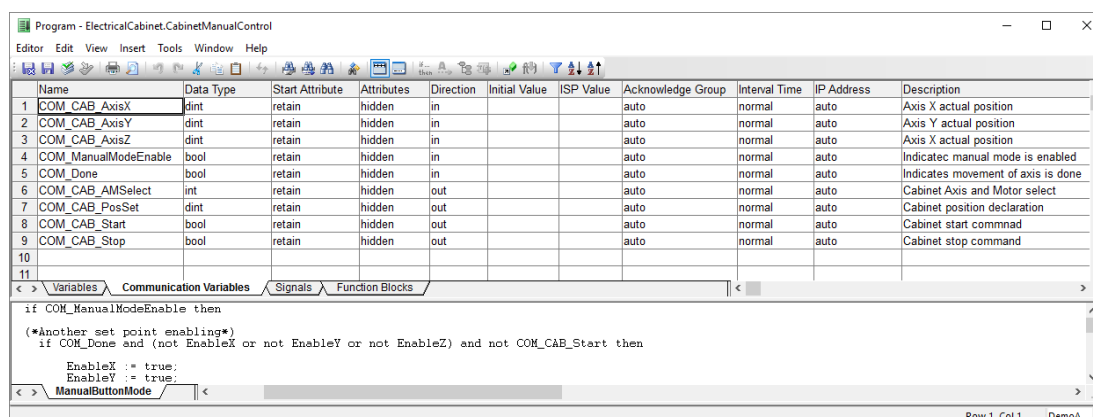
7.1 Proměnné

Deklarace proměnných je realizována pomocí tabulky v horní části jednotlivých programů nebo diagramů a vždy se jedná o lokální proměnné daného bloku. Výjimkou je deklarace globálních proměnných, která je prováděna v souboru aplikace s tím, že tyto proměnné jsou dostupné ze všech programů aplikace. Pokud je potřeba na proměnné navázat signál z I/O karet, musí být nadefinován také příslušný signál v záložce signálů. U proměnných je dále potřeba definovat jejich datový typ a případné startovací a další atributy určující chování proměnné při běhu programu, nebo viditelnost v nadřazených systémech. Dále je možné zadat výchozí hodnotu proměnné, a pokud je propojena se signálem, můžeme vidět také jeho adresu v příslušném I/O modulu. Jako poslední je možné přidat popis.



Obrázek 7.1 Zadávací pole lokálních proměnných

Z důvodu vnitřní logiky některých funkčních bloků, jako Funkčních diagramů, nebo Control Modulů, není možné skrze tyto bloky přistupovat přímo ke globálním proměnným, což znemožňuje komunikace se zbytkem aplikace klasickou cestou. K tomuto účelu se využívá buďto předávání hodnot skrze parametry, nebo pomocí komunikačních proměnných. Komunikační proměnné se kromě jiného využívají pro komunikaci mezi PLC na stejné kontrolní síti. V programech a diagramech, které mají mezi sebou komunikovat, je potřeba deklarovat komunikační proměnné se stejným názvem. Na rozdíl od standardních proměnných u nich musíme navíc nadefinovat několik dalších položek. Těmito parametry jsou rozpoznávací skupina, čas intervalu a IP adresa. Nejdůležitější položkou je pak směr komunikace dané proměnné. V programu, kde je směr toku hodnot nastaven ven, je možné hodnoty měnit. V programu se směrem toku dovnitř lze hodnoty přebírat, ale není možné je měnit. Jedná se tak o jednosměrnou komunikaci, kdy komunikace ven je možná pouze z jedné proměnné s unikátním názvem, kdežto příjem proměnné může být nastaven v neomezeném množství programů a diagramů.



	Name	Data Type	Start Attribute	Attributes	Direction	Initial Value	ISP Value	Acknowledge Group	Interval Time	IP Address	Description
1	COM_CAB_AxisX	dint	retain	hidden	in			auto	normal	auto	Axis X actual position
2	COM_CAB_AxisY	dint	retain	hidden	in			auto	normal	auto	Axis Y actual position
3	COM_CAB_AxisZ	dint	retain	hidden	in			auto	normal	auto	Axis Z actual position
4	COM_ManualModeEnable	bool	retain	hidden	in			auto	normal	auto	Indicates manual mode is enabled
5	COM_Done	bool	retain	hidden	in			auto	normal	auto	Indicates movement of axis is done
6	COM_CAB_AMSselect	int	retain	hidden	out			auto	normal	auto	Cabinet Axis and Motor select
7	COM_CAB_PosSet	dint	retain	hidden	out			auto	normal	auto	Cabinet position declaration
8	COM_CAB_Start	bool	retain	hidden	out			auto	normal	auto	Cabinet start command
9	COM_CAB_Stop	bool	retain	hidden	out			auto	normal	auto	Cabinet stop command
10											
11											

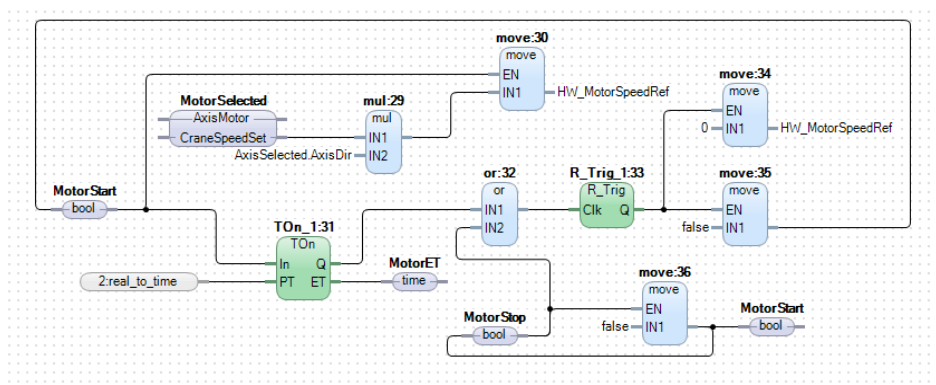

```

if COM_ManualModeEnable then
(*Another set point enabling*)
if COM_Done and (not EnableX or not EnableY or not EnableZ) and not COM_CAB_Start then
    EnableX := true;
    EnableY := true;
  
```

Obrázek 7.2 Deklarace komunikačních proměnných

7.2 Programovací bloky Function Diagram a Control Module

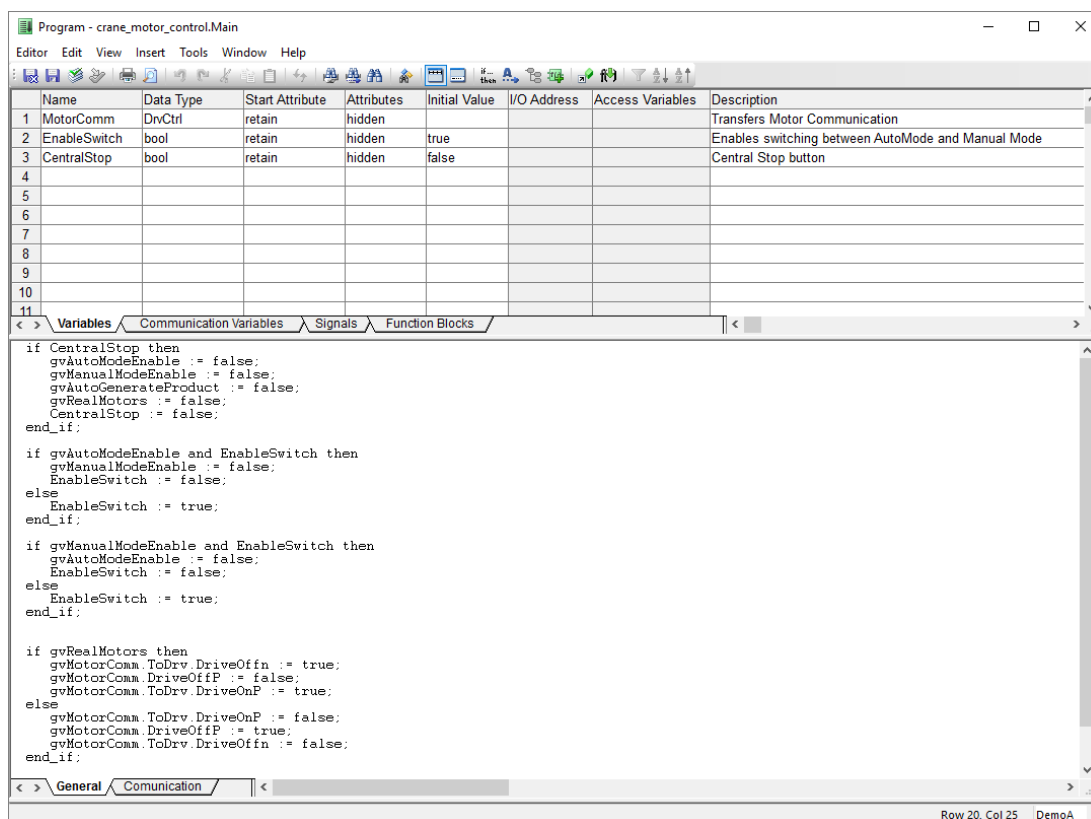
Control module and diagrams are a special type of blocks, which are often used for creating logic of a certain machine or control unit and then they are used by other programs to perform certain tasks. As it was said in the subchapter about variables, access to data of these blocks works in a different mode than for example in programs. Control Modules can be programmed in ST, IL, SFC, FBD and also in LD, they can contain other nested control modules, and it is possible to create their graphical interface for manual control during the application run. Diagrams offer, compared to modules, the possibility of programming in the special programming language ABB Function Diagram, which is a type of programming language used also by other PLC manufacturers, when it comes to the extension of FBD with other expanding functions. Compared to Control Modules, for them it is possible to set the connection of variables to signals.



Obrázek 7.3 Řídící logika vytvořená pomocí funkčního diagramu

7.3 Program

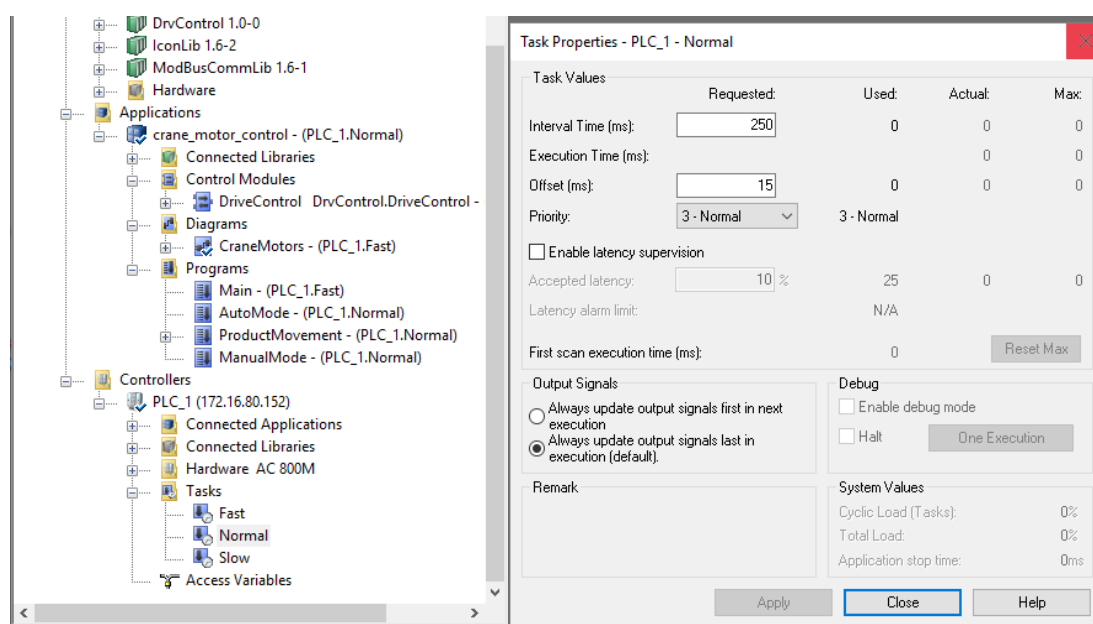
Programy patří k nejvyžívanějším částem aplikace. Mohou být programovány pomocí všech pěti standardizovaných programovacích jazyků a navíc mohou přímo přistupovat ke globálním proměnným dané aplikace. Je možné pro ně nastavit signály a mohou využívat funkční bloky, stejně jako komunikační proměnné. Pomocí programů je řešena většina logických celků v demo úloze.



Obrázek 7.4 Příklad programu programovaného pomocí strukturovaného textu

7.4 Řízení úloh – Task Properties

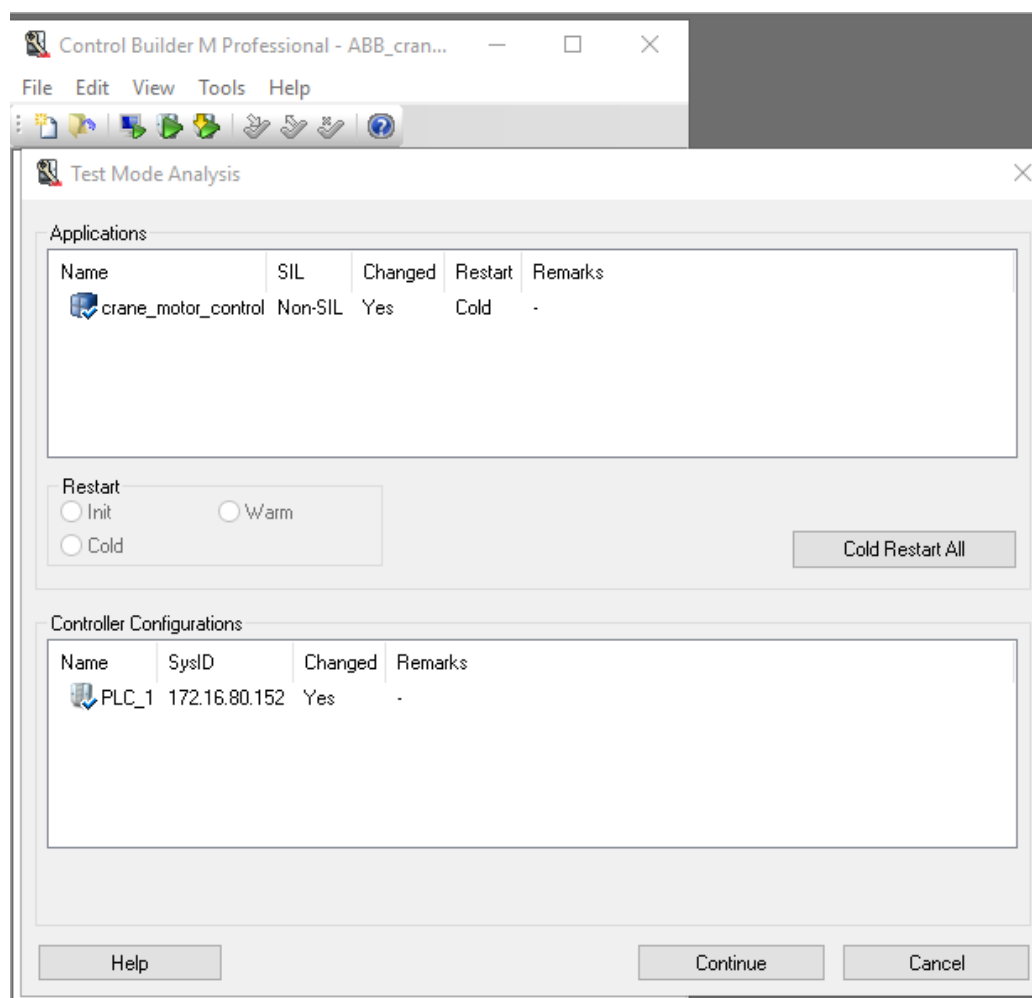
Nedílnou součástí tvorby aplikace je definování Tasků, které je prováděno pro jednotlivé PLC. Tasky jsou časované úlohy, které určují čas a četnost provádění jednotlivých částí aplikace. Tasky je možné nastavit globálně pro celou aplikaci, kdy všechny podřazené položky stromu převezmou task nadřazené položky, nebo je možné dodefinovat tasky pro jednotlivé části aplikace zvlášť. U tasků je možné nastavit čas intervalu, offset, prioritu, a nebo vybrat, zda bude hodnoty proměnných z programů zapisovat do paměti na začátku nebo na konci cyklu. Pro aplikace v demo úloze byly využity výchozí tasky Normal a Fast s intervalem 250, respektive 50 milisekund.



Obrázek 7.5 Nastavení tasků

7.5 Testovací mód, online mód a nahrání programu do PLC

Pro testování, přístup k aplikaci nahrané v PLC a samotnému nahrávání aplikace do kontroléru jsou určeny tři ikony s variacemi symbolu play v horní části Control Builderu. Při nahrávání do PLC je možné zvolit, jaká aplikace bude nahrána do příslušného PLC a je možné ovlivnit, zda při nahrání aplikace dojde k inicializačnímu, studenému nebo teplému restartu, kterýžto výběr ovlivní stav proměnných v kontroléru. Při výběru teplého restartu, pokud to povaha aktualizace aplikace dovolí, nedojde ke ztrátě aktuálních hodnot proměnných. Inicializační nebo studený restart používaný při větších změnách v aplikaci, anuluje hodnoty všech proměnných do výchozího stavu, a proto jej není vhodné využívat v systémech, které běží v ostrém provozu a ztráta hodnot proměnných by mohla způsobit havárii řízené technologie.



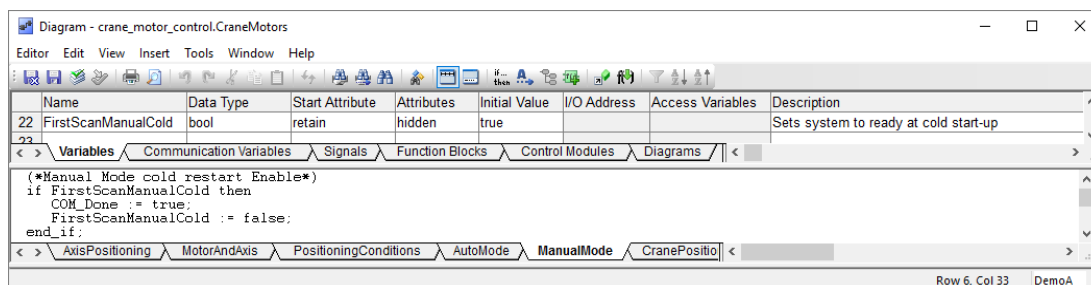
Obrázek 7.6 Volby při spouštění programu v testovacím módu

8 Realizace demo úlohy – Testování a ověření funkce

V okamžiku naprogramování základního kódu bylo nutné otestovat běh aplikace a ošetřit všechny havarijní stavy a chyby v kódu, který by mohli vést k nesprávné funkci aplikace, jejímu zastavení nebo dokonce k poškození připojeného hardwaru. Následující kapitola rozebírá úpravy softwaru a zapojení hardwaru pro zajištění správné funkčnosti aplikace.

Studený start

Při vývoji aplikace je nutné počítat s tím, že testovací mód ve výchozím nastavení funguje na bázi teplého restartu, tudíž přejímá aktuální hodnoty proměnných ve stavu, v jakém byl ukončen poslední běh testovacího módu. Studený restart je proveden pouze pokud je tato možnost vybrána, a nebo je vypnut a znova zapnut celý program Control Builder. Volba studeného restartu je důležitá z pohledu nutnosti nastavit vybraným proměnným výchozí hodnotu při startu systému. To může být dosaženo pomocí definice inicializační hodnoty u vybrané proměnné při její deklaraci, nebo dále v programu pomocí příkazu IF, který je proveden pouze při prvním cyklu programu. Tímto způsobem jsou v aplikaci ošetřeny například hodnoty referenčních otáček jednotlivých simulovaných motorů, určení výchozí polohy jeřábu, nebo zapnutí manuálního módu ihned po startu programu a další. Příslušné ovládací proměnné je vhodné nazvat způsobem, dle kterého je na první pohled patrná jejich funkce, viz obrázek níže a název obsahující sousloví *FirstScan*.

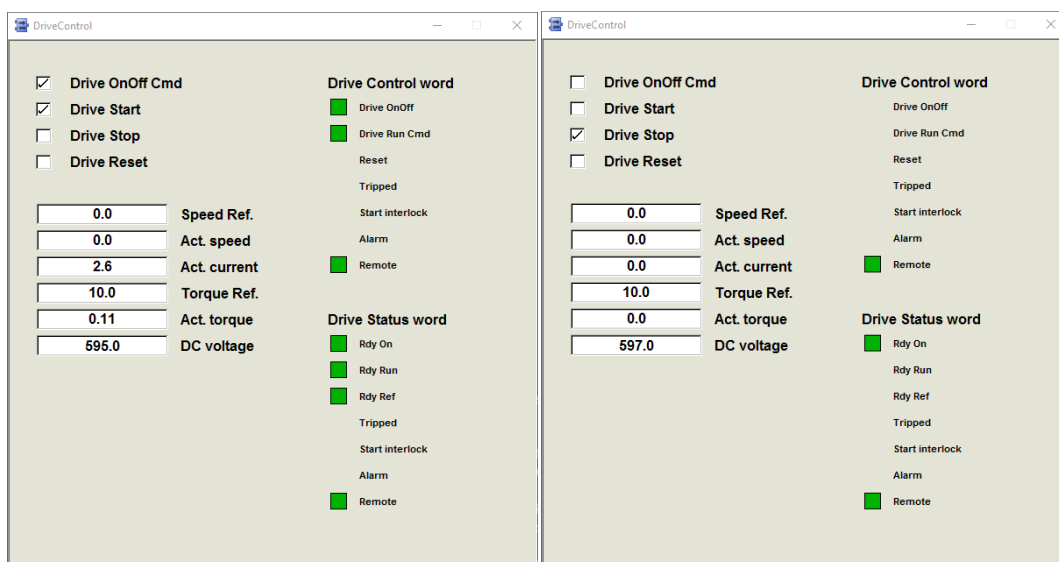


Obrázek 8.1 First Scan podmínky pro nastavení specifických hodnot proměnných při startu aplikace

Ovládání reálných motorů a Central stop

Aplikace ke svému provozu nepotřebuje, aby byl reálný motor zapnut. Zapnutí a vypnutí motoru je možné nastavit pomocí globální proměnné RealMotors, která nastavuje hodnoty kontrolního slova komunikovaného do frekvenčního měniče. Pokud je provoz motoru povolen, je aktivní On/Off Command a příkaz Motor Start. Při tomto nastavení řídí aplikace pohyb motoru pomocí změny referenčních otáček, kdy při kladné referenční hodnotě jede motor v kladném směru otáčení zadanými otáčkami, při nulové referenci je motor zastaven a při zadání záporné reference obrací směr otáčení. Pokud jsou reálné motory zastaveny, dojde k vypnutí příkazu start, zapnutí příkazu stop a následnému vypnutí On/Off Commandu. Vypínací sekvence je spuštěna také v případě, že je aktivováno tlačítko Central Stop.

Dalším bezpečnostním opatřením zavedeným pro skutečný motor bylo nastavení limitu pětiminutového limitu běhu. Po zapnutí fyzického motoru je automaticky spuštěn časovač, který motor po pěti minutách automaticky vypne.



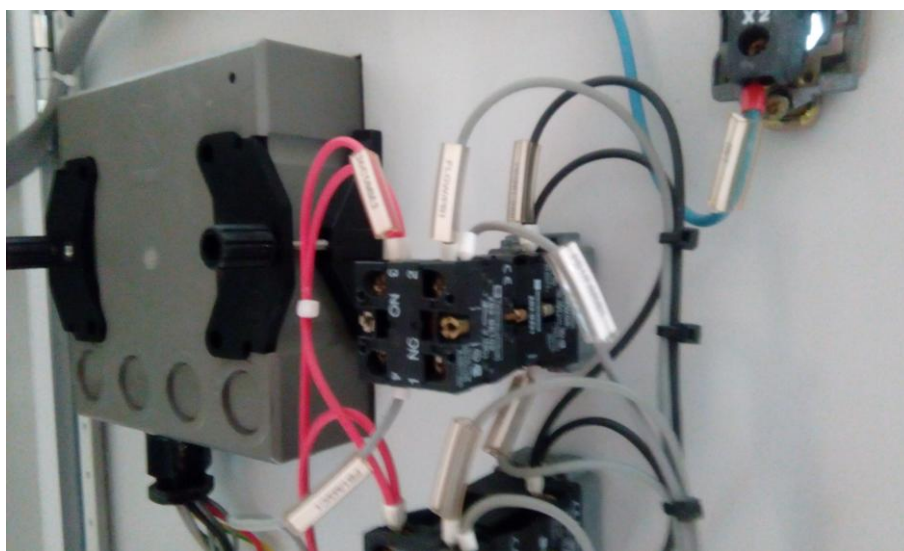
Obrázek 8.2 Command Window Control Modulu při zapnutém a vypnutém reálném motoru

Vypnutí motoru při přepnutí módu

Při testování aplikace bylo nadále zjištěno, že by bylo vhodné při přepínání mezi automatickým a manuálním režimem vypnout motory a zastavit pohyb v příslušné ose. Tento stav byl docílen přidáním podmínky do diagramu Crane Motors, která říká, že pokud není zapnut ani jeden ze dvou provozních módů, pohyb v ose je zastaven a reálnému motoru je nastavena reference nulových otáček. Stejná podmínka byla přidána také při vypnutí ovládání z tlačítkového rozvaděče v manuálním módu.

Přepojení tlačítek v rozvaděči

Při testování řízení jeřábu skrze potenciometry a tlačítka v rozvaděči museli být vyřešeny kromě nutnosti správného nastavení citlivosti přenosu analogových hodnot také problémy s elektrickým zapojením tlačítek, které se projeví až při manipulaci s fyzickými vstupy. Signál z potenciometru byl v předchozích demo úlohách veden po drátu, kterému bylo fyzicky možné rozpojit kontakt při sepnutí příslušného tlačítka. V nové demo úloze byl tento stav nežádoucí, protože program reaguje jak na náběžnou tak padající hranu signálu z tlačítka. Při ztrátě signálu z potenciometru při zamáčknutém tlačítku pak docházelo k nesprávné funkci aplikace, kdy se při rozběhu motoru měnila požadovaná referenční pozice a nedocházelo k zapínání indikačních lamp čekajících na vyrovnaní požadované a aktuální hodnoty. Tento problém byl vyřešen propojením přerušeného kabelu do jedné svorky v tlačítku tak, aby ani po jeho stisknutí nedocházelo ke ztrátě hodnoty.



Obrázek 8.3 Přepojení tlačítek na rozvaděči

Referování při přechodu z manuálního do automatického režimu

Z hlediska bezpečnosti provozu jeřábu v uvažované demo úloze bylo také programově ošetřeno najetí jeřábu do referenční polohy po přepnutí z manuálního do automatického režimu. Manuální režim umožňuje najet jeřábem do libovolného místa ve výrobní hale, a pokud by shodou náhod byl v pozicích mimo cesty určené v automatickém módu, mohlo by dojít například k nabourání do překážek v hale. V daný okamžik také nemusí být jeřáb v jednom z význačných bodů, kdy přítomnost jeřábu v těchto bodech souvisí s provozem automatického módu. Při najetí jeřábu do referenční pozice jsou tak vyřešeny jak záležitosti bezpečnosti, tak bezproblémového přechodu mezi módy. Ke spouštění referenčních sekvencí a přenastavení potřebných proměnných slouží proměnné signalizující, který mód byl naposledy spuštěn. Na základě těchto vlajek jsou pak provedeny příslušné operace v diagramu Crane Motors.

Vypnutí motoru při krátkých přejezdech

Při testování aplikace v režimu ovládání tlačítky a potenciometry docházelo při nastavení krátké vzdálenosti přejezdu k potížím s vypnutím motorů. Problém způsobovala dlouhá komunikační prodleva, kdy podprogram v PLC s rozvaděčem nedostal včas potvrzovací příkaz z PLC s motorem. Chybu kódu, které se při delších přejezdech neprojevovala se podařilo programově ošetřit ve prospěch plné funkčnosti. Kvůli prodlevám komunikace se ruční ovládání jeřábu ukázalo jako nejméně vhodné, oproti ostatním módům je nejméně uživatelsky přívětivý.

Podmínky pro přípravný bod

Jelikož je uživateli demo aplikace umožněno zadávat vlastní názvy výrobků a nastavovat čas jejich výroby, nebo v ručním módu zapisovat požadované polohy jednotlivých os jeřábu, je nutné uživatelem zadaný obsah kontrolovat. Při zadávání produktů k tomuto účelu slouží série podmínek, které hlídají, že není překročena maximální délka názvu deset znaků a k nazvání výrobku není použito zakázané slovo *empty*, prázdný znak nebo název, který je již použit u jiného výrobku dále v procesu. Hlídán je také výrobní čas, který musí spadat do intervalu 1 až 300 vteřin. Pokud nejsou všechny tyto podmínky splněny, není možné výrobek přesunout z přípravného bodu do fronty výrobků k nakládce na jeřáb. Automatické generování výrobků je nastaveno tak, aby všechny výše zmíněné předpoklady splňovalo.

Testovací provoz

Po odstranění všech nedostatků původního kódu byla aplikace důkladně otestována při běhu ve všech dostupných módech a při libovolném přepínání mezi nimi. Hlavním zatěžkávacím testem byl provoz v automatickém módu nejprve s vypnutým a následně zapnutým reálným motorem, následovaném testem Central Stopu a resetovacích tlačítek. Při finálních testováních nebyl nalezen žádný havarijní stav znemožňující další práci aplikace a ta byla prohlášena za otestovanou a funkční.

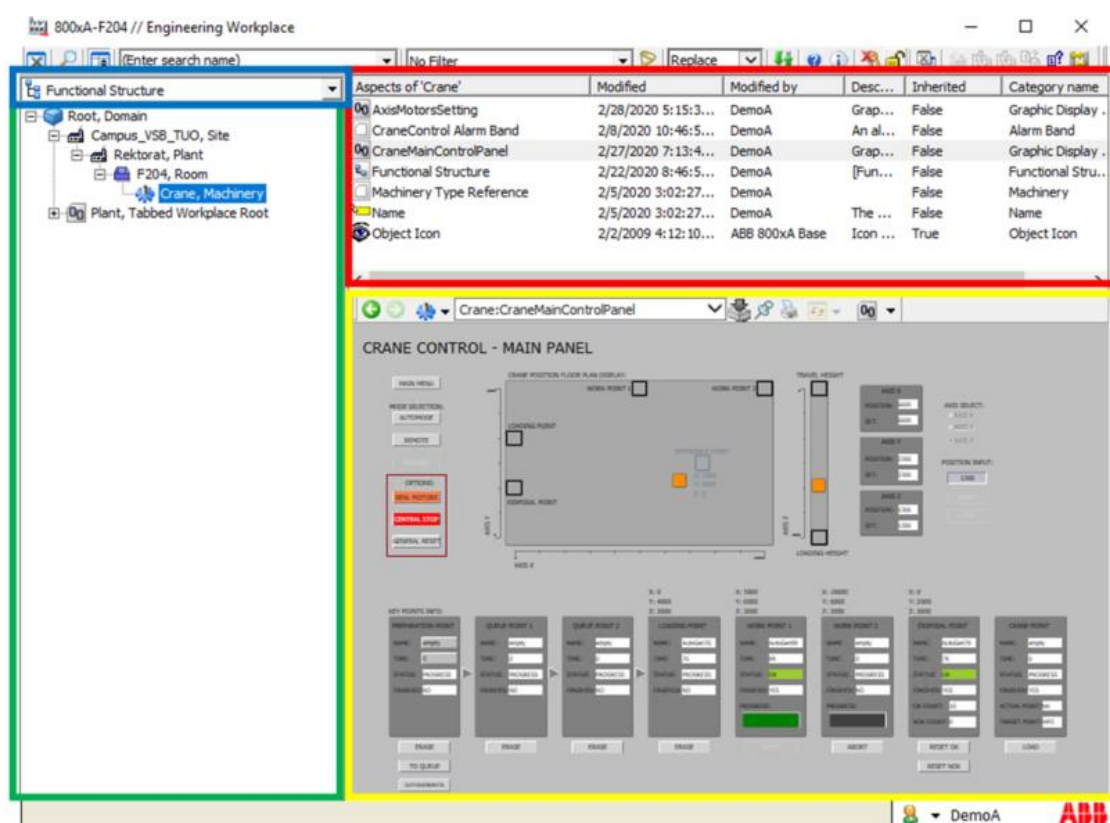
Name	Current Value	Data Type	Start Attribute	Attributes	Initial Value	I/O Address	I/O Description	Access Variables	Description
gvPP		ProductData	retain	hidden					Preparation Point
name	AutoGen35	string	retain		'empty'				Name of product
timer	39	dint	retain		0				time needed to finish produ
status	false	bool	retain		0				ok if finished - nok if aborte
finished	false	bool	retain		0				indicates finished timer or z
gvQP1		ProductData	retain	hidden					Queue Point 1
name	AutoGen23	string	retain		'empty'				Name of product
timer	23	dint	retain		0				time needed to finish produ
status	false	bool	retain		0				ok if finished - nok if aborte
finished	false	bool	retain		0				indicates finished timer or z
gvQP2		ProductData	retain	hidden					Queue Point 2
gvLP		ProductData	retain	hidden					Loadind Point
gvDP		ProductData	retain	hidden					Disposal Point
gvCP		ProductData	retain	hidden					Crane Point
name	AutoGen34	string	retain		'empty'				Name of product
timer	34	dint	retain		0				time needed to finish produ
status	false	bool	retain		0				ok if finished - nok if aborte
finished	false	bool	retain		0				indicates finished timer or z
gvWP1		ProductData	retain	hidden					WorkPoint 1
name	AutoGen48	string	retain		'empty'				Name of product
timer	48	dint	retain		0				time needed to finish produ
status	false	bool	retain		0				ok if finished - nok if aborte
finished	false	bool	retain		0				indicates finished timer or z
gvWP2		ProductData	retain	hidden					WorkPoint 2
gvResetP		ProductData	retain	hidden					Reset Point
gvAutoModeEnable	true	bool	retain	hidden	false				Enables AutoMode
gvCraneInPosition	false	bool	retain	hidden					Indicates crane in Point po
gvManualModeLoad	false	bool	retain	hidden					Manual load/unload in man
gvAutoGenerateProduct	true	bool	retain	hidden					Automatically generates prc
gvRealMotors	false	bool	retain	hidden	false				Enables real motors
gvManualModeEnable	false	bool	retain	hidden	false				Enables ManualMode

Obrázek 8.4 Globální proměnné při běhu aplikace v automatickém módu s vypnutým reálným motorem

9 Základní konfigurace systému 800xA

Systém 800xA je postaven na principu objektově orientovaného programování a je rozdělen do několika struktur, které obsahují objekty, na které jsou navázány aspekty. Konfigurace systému spočívá ve vytváření a úpravě, respektive propojování jednotlivých objektů a aspektů. V minulých kapitolách již bylo nahlíženo do kontrolní struktury obsahující aplikace a připojené PLC kontroléry. Následující kapitola popisuje základní konfigurační kroky předcházející tvorbě operátorských obrazovek.

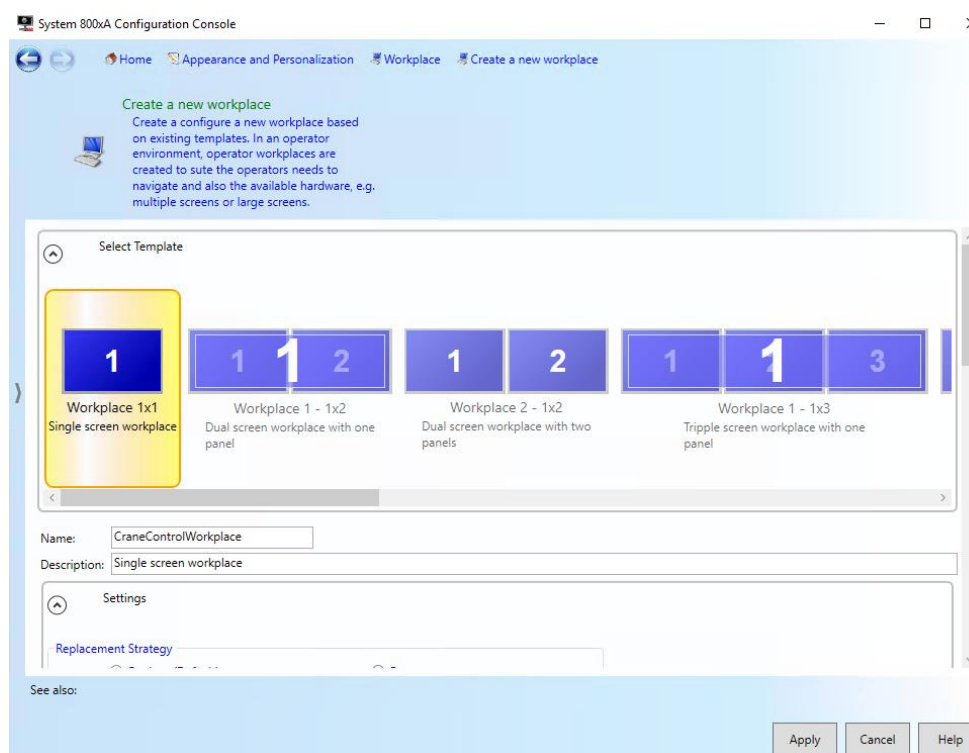
Na níže uvedeném obrázku jsou barevně zvýrazněny jednotlivé části Engineering Workplace. Modře je označena část pro výběr struktury, se kterou chceme pracovat. Zeleně je označena oblast, do které je vypisován seznam objektů v dané struktuře. Po kliknutí na některý z objektů se v červeně ohraničené oblasti zobrazí seznam aspektů náležících danému objektu. Poslední žlutá část poté obsahuje náhled na jednotlivé aspekty, odkud je možné provádět jejich nastavení.



Obrázek 9.1 Přehled obrazovky Engineering Workplace

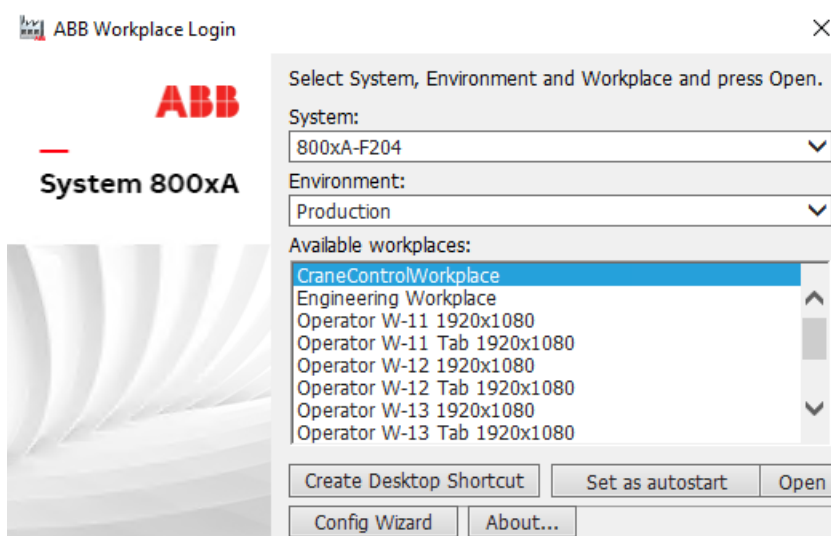
9.1 Vytvoření nového operátorského pracoviště

Jako první bylo pomocí konfigurační konzoly v ABB start menu vytvořeno nové operátorské pracoviště. Po spuštění konfigurační konzoly byla zvolena volba Vzhled a Personalizace -> Pracoviště -> Vytvořit nové pracoviště. Při tvorbě nového pracoviště je kromě názvu možné zvolit další základní parametry vztahující se zejména k použitému zobrazovacímu hardwaru. V případě vytvářené demo aplikace je k dispozici jeden monitor, ale software dokáže pracovat až se čtyřmi monitory v různém vzájemné uspořádání a také dokáže rozlišovat, zda mají být monitory samostatné, nebo pracovat jako jeden větší monitor. Kromě volby zobrazovacího módu, zadání jména a případně popisu pracoviště je dále možné zvolit strategii překrývání obrazovek a základní obsah obrazovky. V horní části obrazovky je možné povolit Application Bar a nastavit jeho výchozí obsah, stejně jako výšku. Pro spodní část obrazovky existuje obdobná volba, oblast obrazovky se jmenuje Status Bar. Application a Status bar mají tu vlastnost, že jsou připnuty k pracovišti a jsou stále dostupné i při přechodu mezi různými obrazovkami.[35]

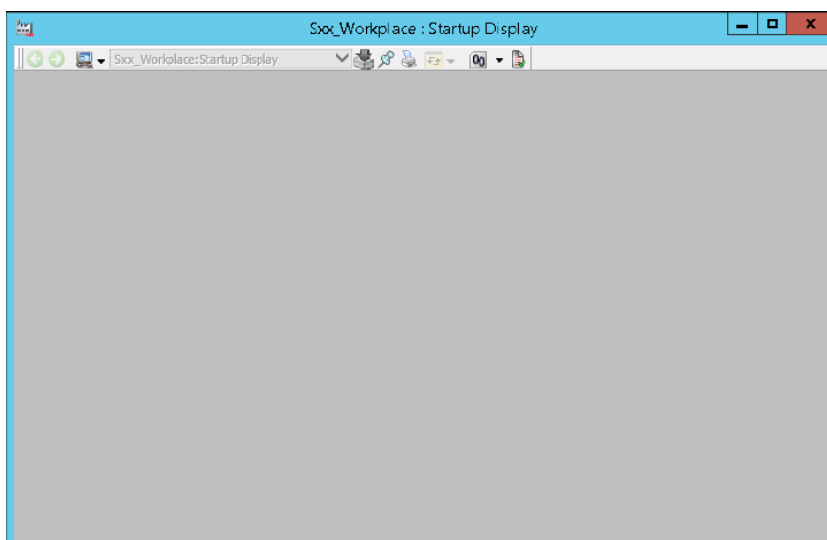


Obrázek 9.2 Konfigurace nového operátorského pracoviště

Všechna výše uvedená nastavení jsou později dostupná ze struktury systému Workplace Structure. Po vyplnění základních parametrů byly úpravy potvrzeny a nově vytvořené pracovní prostředí bylo zpřístupněno ve spouštěcí konzole. Pracoviště je možné spustit stejně jako Engineering Workplace pomocí aplikace Workplace z ABB Start Menu. Tímto krokem se stalo operátorské pracoviště dostupné ze všech stanic v distribuovaném systému. Spouštěcí konzola navíc nabízí automatické spouštění vybraného pracoviště, nabízí přístup do konfigurace systému a dovoluje k vybranému pracovišti vytvořit odkaz na ploše pracovní stanice. Po spuštění nově vytvořeného pracoviště je uživateli k dispozici pouze prázdná startovací obrazovka s ikonami pro navigaci mezi obrazovkami a ikonami pro připnutí a tisk pracovní obrazovky. [35]



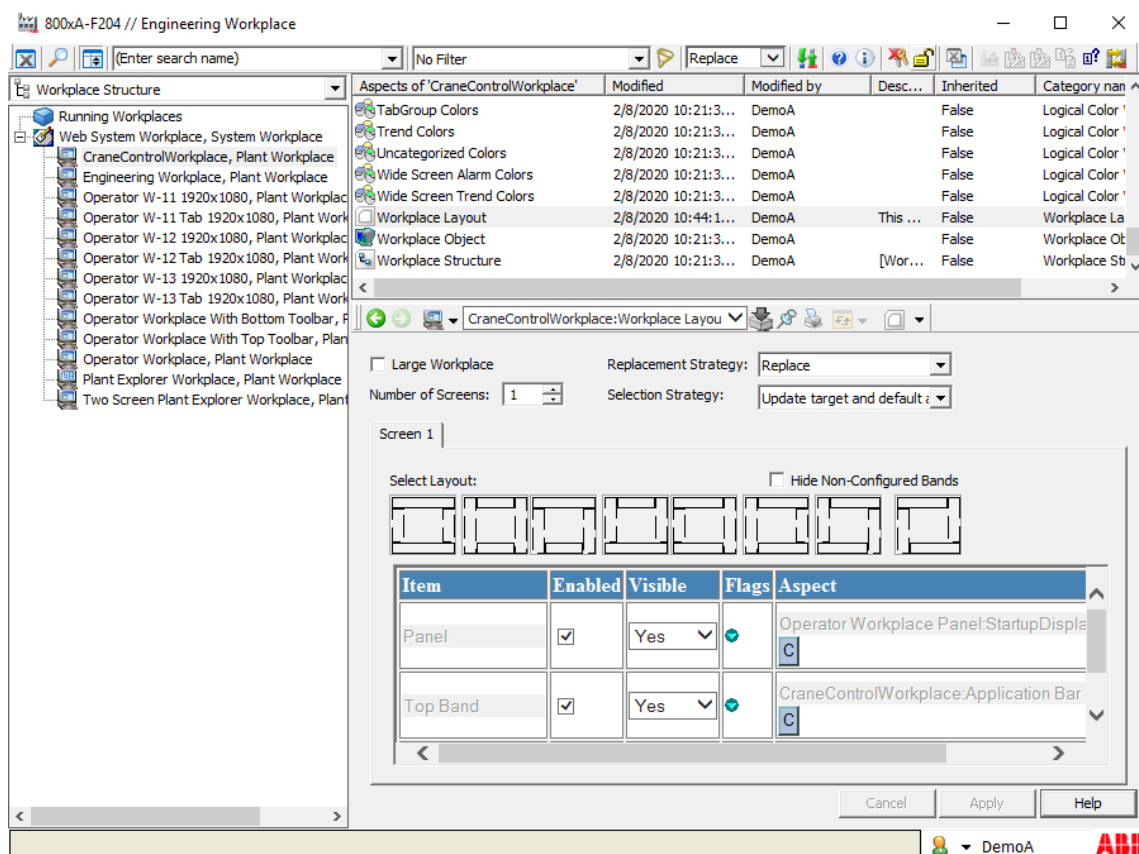
Obrázek 9.3 Výběr spouštěného pracoviště



Obrázek 9.4 Prázdné operátorské pracoviště

9.2 Konfigurace operátorského pracoviště

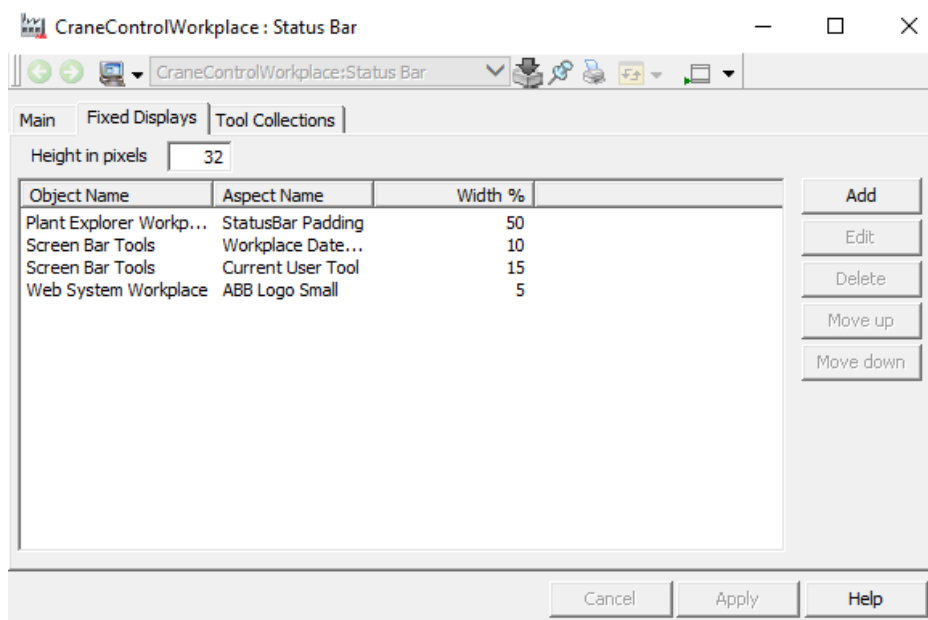
Po úspěšném vytvoření nového operátorského pracoviště bylo přikročeno k jeho další konfiguraci. Úpravy byly prováděny na inženýrském pracovišti spustitelného z konzoly Workplace (postup spuštění popsán v předchozích kapitolách). Ve struktuře je již viditelné nově vytvořené pracoviště. K pracovišti je možné přidávat nové objekty a aspekty, stejně jako prohlížet a upravovat již automaticky vytvořené. Kromě jiného se pod pracovištěm nachází položka *Startup Display*, kterou je možné editovat, jak bude popsáno v následujících kapitolách. Při kliknutí na jednotlivé položky pracoviště pravým tlačítkem myši a volbě *Config View* se spustí konfigurační okno umožňující úpravy daného aspektu. K vlastnostem aspektu je možné přistoupit také z okna náhledu pod seznamem aspektů. Na obrázku níže vybrán *Workplace Layout*, ve kterém je možné změnit základní nastavení provedená při inicializaci pracoviště popsané v minulé podkapitole.[35]



Obrázek 9.5 Workplace Structure

9.4 Nastavení aplikačního panelu a status panelu

K důležitým oblastem pracovního prostředí patří horní a spodní panel. Obě součásti pracoviště jsou uživateli neustále dostupné. V Aplikačním panelu mohou být umístěny například hlášení alarmů nebo jiné informace o systému. V případě demo aplikace není aplikační panel využíván. Naopak spodní status panel využíván je a obsahuje několik základních funkcionalit. Po zvolení *Config View* u aspektu Status Bar je v záložce *Main* je možné přenastavit volby vygenerované při inicializaci pracoviště. Pod záložkou *Fixed Displays* se pak nabízí možnost vložení jednotlivých složek panelu, jejich šířka a celková výška panelu. Pro demo aplikaci byly do Status Baru přidány kromě jiného také hodiny zobrazující čas a datum. Stejně jako u horního panelu je možné do spodního panelu vložit libovolné aspekty. Pod záložkou *Tool Collections* je pak možné do panelu vložit další nástroje pro operátorské pracoviště.[35]

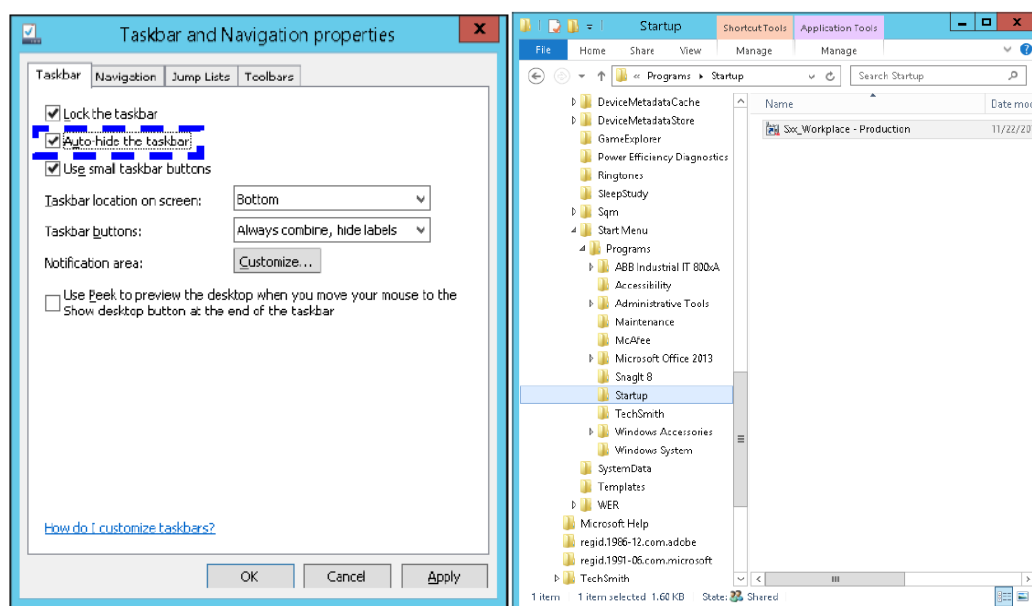


Obrázek 9.6 Nastavení Status Bar

9.5 Bezpečnostní nastavení

Jelikož jsou operátorská pracoviště často určeny pro provoz s velkým důrazem na bezpečnost práce, obvykle bývá vyžadováno, aby operátor nebyl schopen na pracovní stanici opustit aplikaci pracovního prostředí. Operátor se tradičně přihlašuje na účet Windows, který spravuje systémový administrátor. Z toho důvodu je možné nastavovat všechna bezpečnostní opatření pro jednotlivé uživatele systému stejně jako při správě jakékoliv jiné podnikové sítě a pracovních stanic. Pro operátorské pracoviště může být důležité zejména zamezit přístupu na pracovní plochu. Z tohoto důvodu je nutné, aby operátorské pracoviště splňovalo následující podmínky. Při přihlášení uživatele musí být automaticky spuštěno operátorské pracoviště a musí být zobrazena příslušná úvodní obrazovka. Operátorské pracoviště musí běžet v módu celé obrazovky, nesmí být viditelná spodní lišta s nabídkou start ani nesmí být možné ji vyvolat. [35]

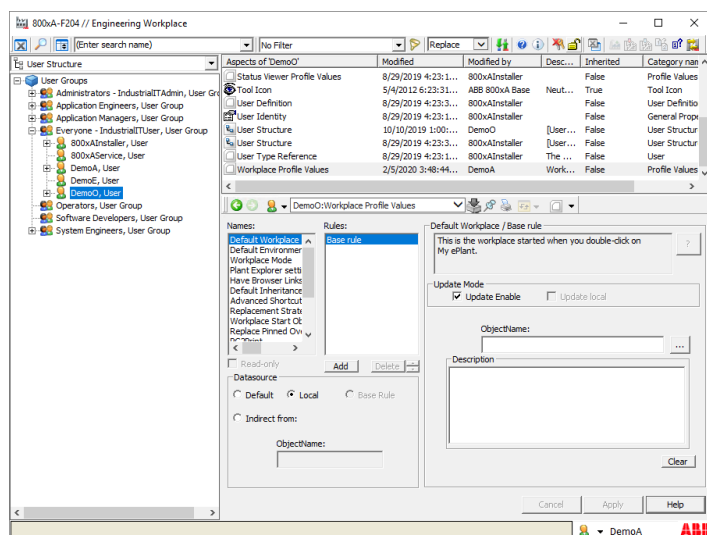
Pro automatické spuštění Operátorského pracoviště je možné ze spouštěcí konzoly vygenerovat zástupce na plochu a toho následně umístit do systémové složky Auto Startup. Poté se bude pracoviště automaticky spouštět s přihlášením uživatele do systému. [35]



Obrázek 9.7 Nastavení v systému Windows

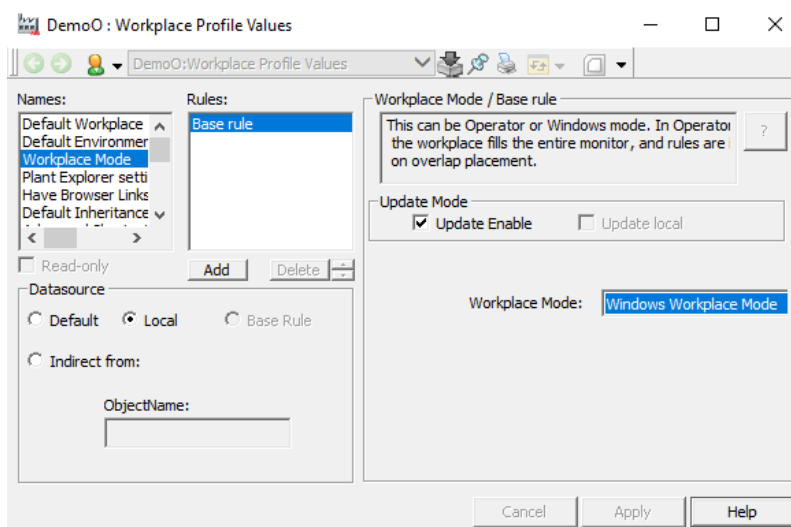
Nastavení pro uživatele systému 800xA

Nastavení výchozího operátorského pracoviště stejně jako další nastavení byla provedena v uživatelské struktuře User Structure, která shrnuje uživatele zavedené do systému 800xA. Pod položkou Everyone je možné přistoupit ke všem uživatelům v systému. Po kliknutí na daného uživatele je nutné vybrat aspekt Workplace Profile Values, odkud je možné upravovat výchozí nastavení pro jednotlivé uživatele. [35]



Obrázek 9.8 Nastavení systému pro jednotlivé uživatele

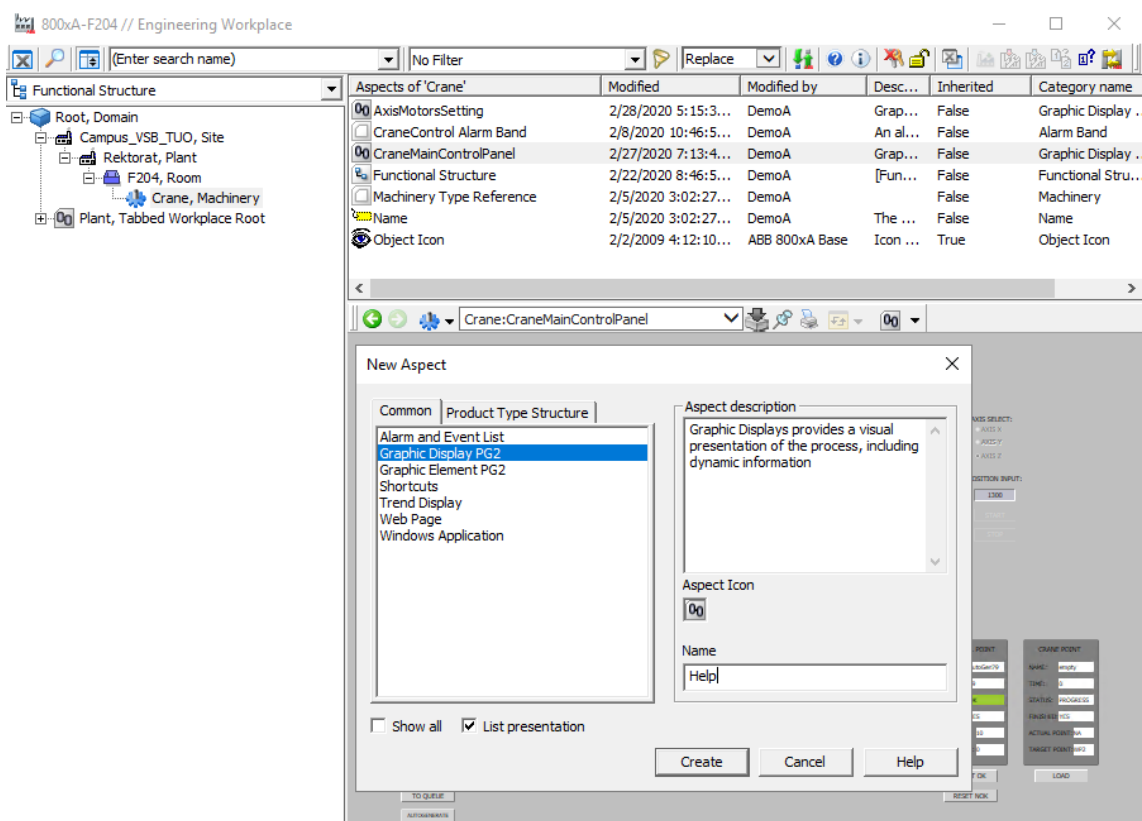
V případě demo úlohy byla operátorská stanice pro uživatele DemoO nakonfigurována následujícím způsobem. Jako první bylo nastaveno výchozí operátorské pracoviště, do prázdné kolony Object Name bylo nutné vybrat pracoviště z příslušné struktury, v případě demo aplikace CraneControlWorkplace. Jako další bylo v nabídce Workplace Mode vybráno nastavení Operator Workplace Mode, které nedovoluje překrytí operátorského pracoviště jinými aplikacemi a hláškami Windows. [35]



Obrázek 9.9 Nastavení módu aplikace

10 Funkční struktura a program Graphics Builder

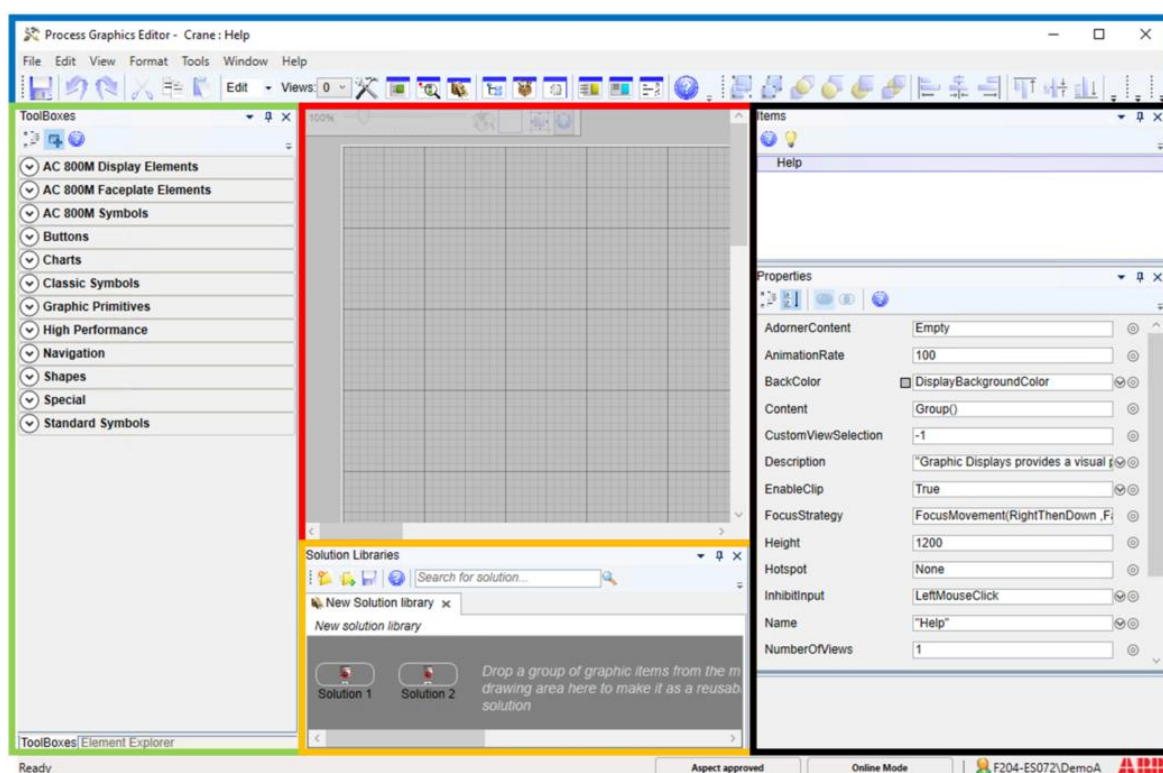
Po dokončení konfigurace operátorského pracoviště mohlo být přistoupeno k tvorbě operátorských obrazovek. Pro následující práci je klíčová funkční struktura systému (Function Structure) obsahuje zejména objekty, ze kterých je možné vytvořit logickou strukturu výrobního podniku. Díky symbolickým objektům, jako oblast (site), továrna (plant), místnost (room) a případně strojní vybavení (machinery) je možné provázat jednotlivé objekty tak, aby odpovídali fyzické skladbě objektů a byly pro operátory i inženýry pracující v systému přehledné. K jednotlivým objektům je možné připojovat jak další objekty, tak aspekty. Jedním z aspektů, který je možné k objektům přidat je také Graphic Display PG2 – operátorskou obrazovku. Display je možné libovolně pojmenovat a následně vytvořit a upravovat. Po kliknutí pravým tlačítkem na daný aspekt a výběrem možnosti *Edit* se automaticky spustí prostředí pro tvorbu obrazovek – Graphics Builder. [35]



Obrázek 10.1 Funkční struktura s objekty operátorských obrazovek

10.1 Základní přehled programu Graphics Builder

Na obrázku níže jsou zvýrazněny jednotlivé části programu, kde v horní části obrazovky ohraničené modrým rámečkem je základní pás karet. Zelené ohraničení patří toolboxu, ze kterého je možné vkládat objekty na plochu operátorské obrazovky, ohraničené červeně. Oranžový rámeček značí oblast solution library, kam je možné ukládat různé celky vytvořené na ploše. Poslední oblast označená černě nese název properties a jsou v ní zobrazovány detailní informace k jednotlivým objektům vloženým na pracovní plochu. Kapitola dále obsahuje přehled základních funkcionalit programu využívaných při tvorbě obrazovek.[35]



Obrázek 10.2 Přehled programu Graphics Editor

10.2 Aktivní prvky

Aktivní prvky slouží k přímé interakci mezi uživatelem a systémem, kdy pomocí tlačítek, posuvníků a dalších ovládacích prvků může uživatel nastavovat hodnoty proměnných. Na obrázku níže je výčet nejpoužívanějších prvků. Vpravo nahoře jsou ukázány tzv. Radio Buttons, které je možné sloučit do skupin a následně pomocí nich definovat jednoznačnou hodnotu proměnné v čase. Všechny Radio Buttons z dané skupiny mají přiřazeny jednu výstupní proměnnou, do které je následně zapsána hodnota konkrétní volby. V pravém horním rohu jsou k vidění Toggle Buttons, které po stisknutí drží novou pozici, stejně jako možnost zneplatnění tlačítka, kdy je na základě stavu systémových proměnných možné zakázat tlačítkům interakci. Ve spodní levé části obrázku je vidět hned několik dalších typů vstupních objektů, jednak Push Button s názvem Erase, který po stisku zapíše hodnotu a navrátí se do původní polohy, tak vstupní pole pro zápis čísel nebo textu se zvýrazněným rámečkem. Jak je z obrázku dále patrné, k jednotlivým objektům na pracovní ploše je možné zapsat nápovědu, která se zobrazí po najetí ukazatele myši nad objekt. Na obrázku v pravém dolním rohu je poté vidět několik dalších funkcionalit, kdy na základě hodnoty proměnné může být měněn obsah jednotlivých polí, případně mohou být pole podbarvována, nebo je využito dalších objektů, jako status bar pro zobrazení postupu určité operace v závislosti na čase. V neposlední řadě stojí také textová pole, která slouží pro zobrazení hodnot ze systému. Mezi často používané prvky patří také zobrazovače grafů a trendů, stejně jako ikony prvků používaných v procesní technice.[35]

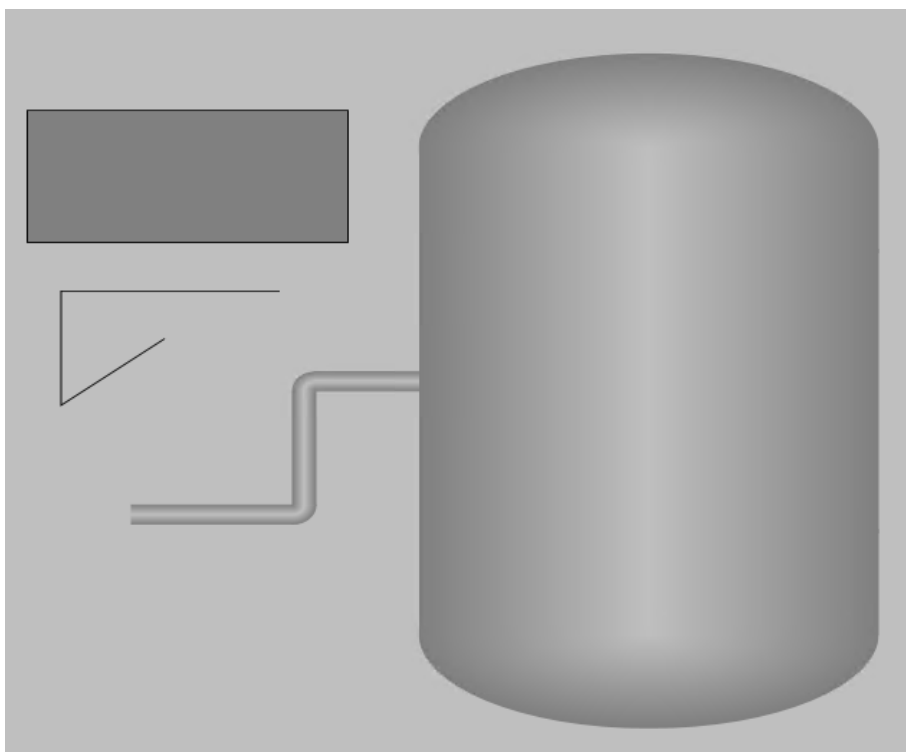
The screenshot displays a control interface with several sections:

- MOTORS PARAMETERS:** Contains two radio buttons: "DEFAULT SETTING" (selected) and "EDITED SETTING".
- MODE SELECTION:** Contains three buttons: "AUTOMODE", "REMOTE", and "MANUAL".
- KEY POINTS INFO:** A section containing two panels:
 - PREPARATION POINT:** Includes text fields for "NAME" (value: empty), "TIME" (value: 0), "STATUS" (value: PROGRESS), and "FINISHED" (value: NO). A tooltip "Input name" is visible next to the TIME field. Below the fields is an "ERASE" button.
 - WORK POINT 1:** Includes text fields for "NAME" (value: AutoGen89), "TIME" (value: 89), "STATUS" (value: OK, highlighted in green), and "FINISHED" (value: YES). Below these is a "PROGRESS" section with a green progress bar.

Obrázek 10.3 Výběr aktivních prvků - tlačítka a textová pole

10.3 Pasivní prvky

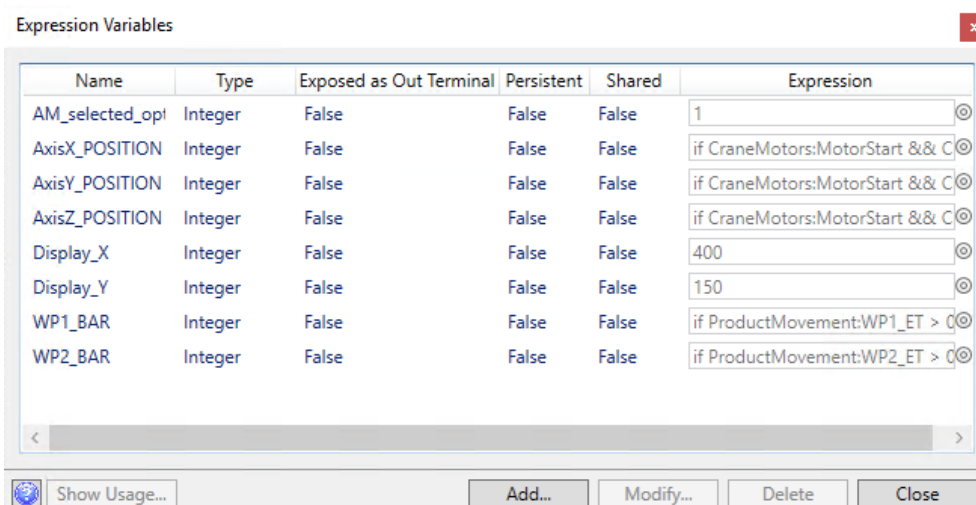
Mezi pasivní prvky patří zejména textové a tvarové prvky, které nejsou určeny pro zobrazování nebo nastavování hodnot přenášovaných do kontrolní struktury systému. Zároveň je možné chování těchto prvků upravovat na základě proměnných kontrolní struktury nebo proměnných vytvořených pomocí zápisu do tabulky Expression Variables. Prvky slouží zejména pro tvorbu vizuálního prostředí, zpřehlednění obrazovek, ohraničení důležitých oblastí a jsou důležitou částí uživatelských obrazovek, kdy je na základě těchto prvků možné zpřehlednit operátorům orientaci v daném procesu. Na obrázku níže jsou k vidění některé základní a často používané prvky, jako prvek tvaru, čáry, potrubí a nádrže. K pasivním prvkům dále mohou patřit vložené fotografie či nákresy.[35]



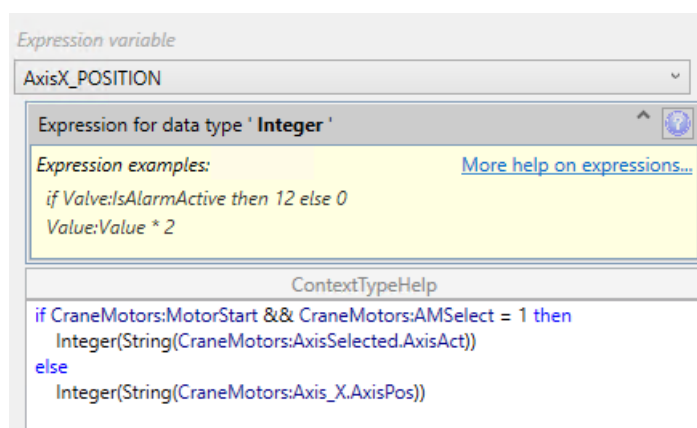
Obrázek 10.4 Příklad několika pasivních prvků

10.4 Výrazy a proměnné

Velkou předností programu je editor proměnných, ve kterém je možné vytvořit nové proměnné využívané zejména pro potřeby dynamické vizualizace. Kromě možnosti zadání nových proměnných a pevné definice jejich hodnot, je důležitá také možnost zapsání logického výrazu, který může pracovat se skutečnými hodnotami čerpanými z kontrolní struktury. Tímto způsobem je možné dále zpracovávat surová data z PLC, které tak nemusí řešit složitější výpočty, ale přenechá je výkonnějšímu serveru s výrazně větším výpočetním výkonem. Tento předpoklad však neplatí vždy a je důležité promyslet, na jaké úrovni data zpracovat, aby nedošlo k zahlcení serveru pomocnými výpočty. Druhý obrázek pak ukazuje pole pro zápis výrazu. Syntaxe zápisu je intuitivním hybridem mezi Structed Text a jazykem C. Automaticky je také zobrazována nápověda.[35]



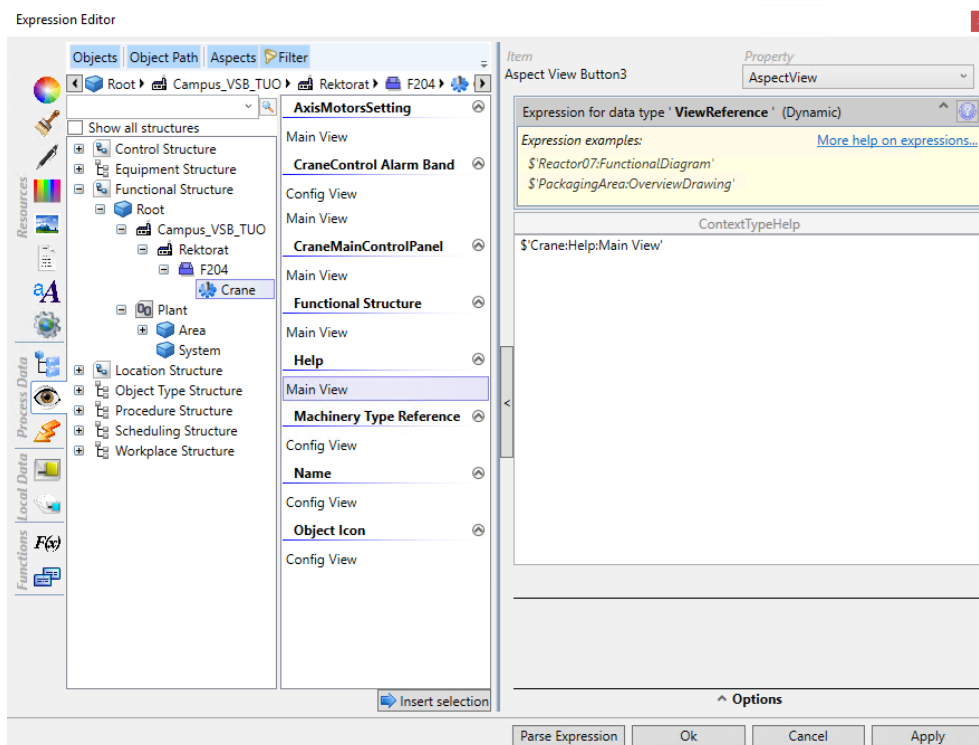
Obrázek 10.5 Vytvoření nových výrazů a proměnných



Obrázek 10.6 Zápis logického výrazu v Expression Editoru

10.5 Odkaz na aspekty

V okamžiku, kdy je potřeba vytvořit odkaz na jinou operátorskou obrazovku, využívá se tlačítek z toolboxu s názvem Aspect View Button. Po vytvoření tlačítka je možné ho upravit stejně jako klasická tlačítka. Hlavním rozdílem je volba Aspect View, kde po rozkliknutí Expression Editoru byly vybrány příslušné struktury, ve kterých jsou umístěny aspekty operátorských obrazovek. [35]



Obrázek 10.7 Nastavení parametru Aspect View pro odkaz na obrazovku help

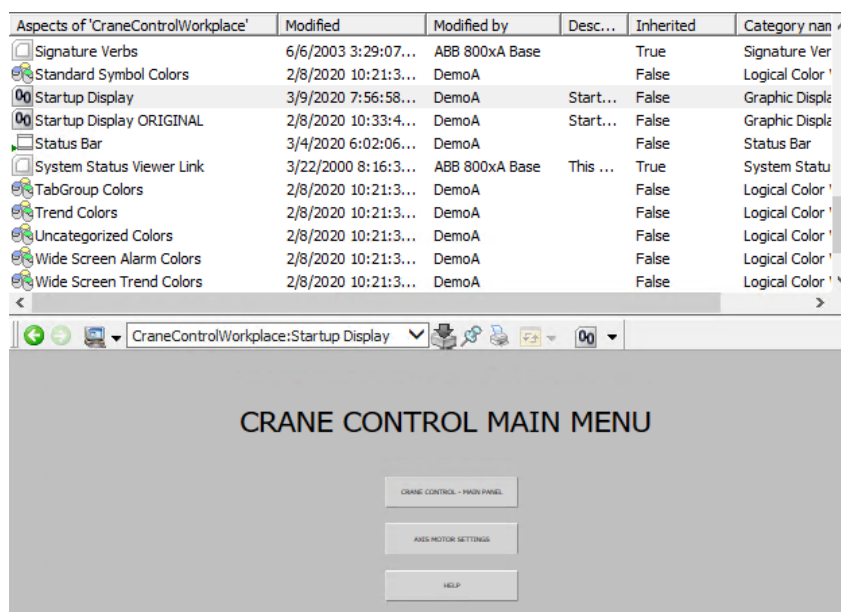
Kromě výše uvedeného výběru prvků obsahuje systém podporu například ActiveX technologie, možnosti vložení živého videa, pokročilé využívání skriptů a doinstalování specifických knihoven vizualizačních objektů profilovaných zejména do oblastí častého nasazení systému 800xA, jako je ropný a chemický průmysl. Z celkového pohledu je program Graphics Builder velmi pokročilým nástrojem pro tvorbu operátorských obrazovek a díky přehlednému a uživatelsky přívětivému pracovnímu prostředí je s ním práce rychlá a snadná. Napomáhá k tomu také skutečnost, že díky přímé integraci do systému 800xA disponuje snadným přístupem k datům kontrolních aplikací na PLC, kdy u jednotlivých proměnných v PLC stačí pouze zrušit atribut Hidden, který slouží právě k schování proměnné před systémem, a daná proměnná je skrze OPC server ihned dostupná v Graphics Builderu. [35]

11 Vizualizační obrazovky Demo úlohy pro operátorskou stanici

Na základě grafického návrhu funkční specifikace bylo rozhodnuto o vytvoření celkem čtyř vizualizačních obrazovek pro operátorskou stanici. První obrazovka s názvem Startup Display byla vybrána jako rozcestník pro přístup k dalším obrazovkám a plní funkci hlavního menu. Do hlavního menu byly vloženy odkazy na tři další obrazovky. Na obrazovka Crane Control – Main Panel byly zakomponovány všechny ovládací a zobrazovací prvky nezbytné pro obsluhu demo aplikace. Třetí obrazovka s názvem Axis Motor Setting obsahuje nastavení jednotlivých motorů, převodových poměrů a dalších specifikací určujících rychlost pohybu v osách. Poslední obrazovka pak byla přizpůsobena do podoby nápovědy k funkci aplikace i jednotlivým ovládacím prvkům.

11.1 Hlavní menu

Obrazovka hlavního menu byla nastavena jako výchozí obrazovka pro operátorské pracoviště CraneControl Workplace. Obrazovka obsahuje odkazy na další obrazovky v systému. V zásadě se jedná o nejjednodušší obrazovku celé demo úlohy, která slouží pouze jako rozcestník pro přístup k důležitějším obrazovkám. Tlačítka pro přechod na další obrazovku jsou vytvořeny jako odkazy na aspekty a stejně jako ve zbytku demo úlohy je u nich nastavena strategie překreslení obrazovky ve formátu nahrazení, kdy je původní obrazovka zavřena a nahrazena novou obrazovkou. Na obrazovku hlavního menu je možné se ze všech ostatních obrazovek vrátit pomocí tlačítka Main Menu. Na obrázku níže je k vidění struktura operátorského pracoviště s aspektem Startup Display, který je automaticky spouštěn při startu. O řádek níže je k vidění také zálohovaný původní prázdný aspekt. V náhledu na spodní straně obrázku jednoduché hlavní menu.



Obrázek 11.1 Detail pracoviště ve struktuře Workplace s náhledem na obrazovku

11.2 Kontrola jeřábu – hlavní panel

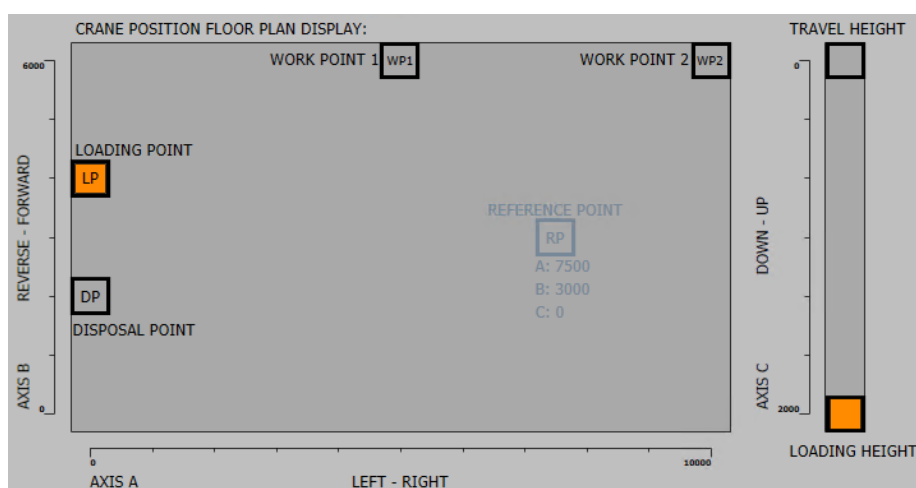
Hlavní obrazovka aplikace byla navržena tak, aby obsahovala všechny informace nutné k bezpečnému provozu demo úlohy. Do levé části obrazovky byla umístěna tlačítka pro navigaci mezi jednotlivými režimy demo úlohy. Jak již vyplynulo z programové definice jednotlivých módů v kapitole zabývající se tvorbou aplikace pro PLC, demo úlohu je možné provozovat v jednom ze dvou režimů – automatickém (Automode) a vzdáleném (Remote – ovládání přímým zadáním na obrazovce nebo panelu). Základní obrazovka obsahuje podřízené ovládací prvky, které byly zpřístupněny na základě volby režimu. Pro operátora to znamená, že rozšiřující funkce vzdáleného a ručního režimu jsou dostupné pouze při volbě tohoto módu. Manuální mód (ovládání z rozvaděče) je dostupný pouze jako podmnožina vzdáleného módu.

Pod tlačítky pro výběr módu se nachází panel důležitých nastavení. První tlačítko slouží k zapnutí a vypnutí reálných motorů a stavový řádek pod tlačítkem vypisuje současný stav motorů. Dále je dostupné tlačítko Cental Stop, které po stisknutí okamžitě zastaví všechny probíhající procesy a také motor. Poslední tlačítko slouží k úplnému restartu celé demo úlohy.

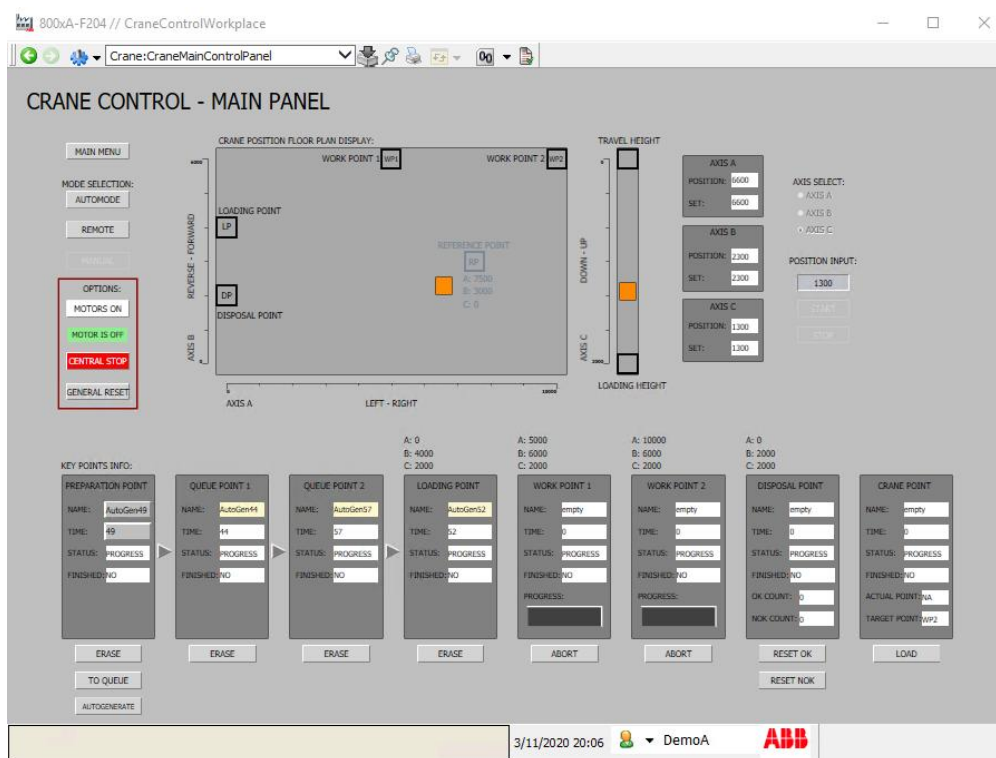
Section	Item	Value / State		
MODE SELECTION	AUTOMODE	Button		
	REMOTE	Button		
	MANUAL	Button		
	OPTIONS	MOTORS ON, MOTOR IS OFF, CENTRAL STOP, GENERAL RESET	Buttons	
AXIS A	POSITION	0		
	SET	0		
	AXIS B	POSITION	4000	
		SET	4000	
		AXIS C	POSITION	2000
			SET	1300
AXIS SELECT			AXIS A	Radio button
			AXIS B	Radio button
	AXIS C		Selected radio button	
POSITION INPUT	Input field		1300	
	START	Button		
	STOP	Button		

Obrázek 11.2 Hlavní volby a poziční tabulka os s ručním zadáváním

Napravo od důležitých tlačítek se nachází mapa výrobní haly s vyznačenými hlavními uzlovými body. První větší mapa představuje půdorysný pohled a zobrazuje aktuální polohu jeřábu v podobě oranžového čtverce. Druhé pomocné zobrazení ukazuje aktuální vertikální polohu jeřábového háku. Vedle polohových obrazovek jsou umístěny panely ukazující polohu jednotlivých os. Osa A slouží pro pohyb doleva a doprava, osa B pro pohyb vpřed a vzad a osa C pro pohyb nahoru a dolů. U jednotlivých os je vidět současná pozice osy a nastavení požadované pozice. Pokud je aktivován vzdálený režim, je možné volit mezi ovládanými osami, ručně zadávat požadované pozice a pohyb jeřábu spouštět a zastavovat pomocí tlačítek start a stop.



Obrázek 11.3 Detailní pohled na zobrazení polohy jeřábu



Obrázek 11.4 Hlavní operátorská obrazovka – celkový pohled

Spodní část obrazovky vyplňují přehledové panely pro jednotlivé klíčové body demo úlohy, nad kterými byly umístěny jejich souřadnice v jednotlivých osách. Všechny panely obsahují celkem čtyři parametry. Jako první je zobrazován název výrobku v daném bodu, který je v případě přítomnosti výrobku pro větší přehlednost podsvícen. Jako druhý parametr je zobrazován zadaný výrobní čas. Parametr status zobrazuje tři různé stavy. Pokud se na daném bodě nachází výrobek, který ještě neprošel výrobním bodem, vypisuje se hodnota „PROGRESS.“ V okamžiku, kdy výrobek dorazí na jeden z výrobních bodů, a dojde k úspěšnému dokončení výrobní operace, dostane značku OK, v opačném případě, kdy je před dokončením výroby zmáčknuto přerušovací tlačítko, dostane značku NOK. Poslední základní parametr značí, zda byl výrobek dokončen nebo nedokončen. Pod panely jsou umístěny tlačítka rozšiřujících funkcí. U přípravného bodu je možné zadat název výrobku a jeho výrobní čas kliknutím na příslušná políčka přímo v panelu. Výrobek setrvá na přípravném bodu až do chvíle, kdy operátor klikne na tlačítko „do řady,“ které posune výrobek do řady končící u nakládacího bodu. Tlačítko Autogenerate spouští automatické generování výrobků, které jsou zároveň automaticky posouvány do výrobní řady. Všechny čtyři první body disponují tlačítkem pro smazání výrobku z daného bodu. Výrobní body jsou rozšířeny o status bar zobrazující postup výroby a namísto tlačítka vymazat disponují tlačítkem zastavit výrobu. Odkládací bod je naopak vybaven počítadlem dobrých a špatných kusů a tlačítka pro reset jednotlivých počítadel. Jeřábový bod disponuje rozšířeným zobrazením, kdy kromě standardních informací je možné odečíst také současnou pozici (pokud se nachází v jednom z klíčových bodů) a také další cílovou pozici v případě provozu v automatickém režimu. Jeřábový bod má k dispozici tlačítko, dostupné pouze při provozu ve vzdáleném nebo manuálním režimu, sloužící k ručnímu přesunu výrobků mezi klíčovými body. K přesunu dojde pouze v okamžiku, kdy jeřáb najede na přesnou adresu klíčového bodu.

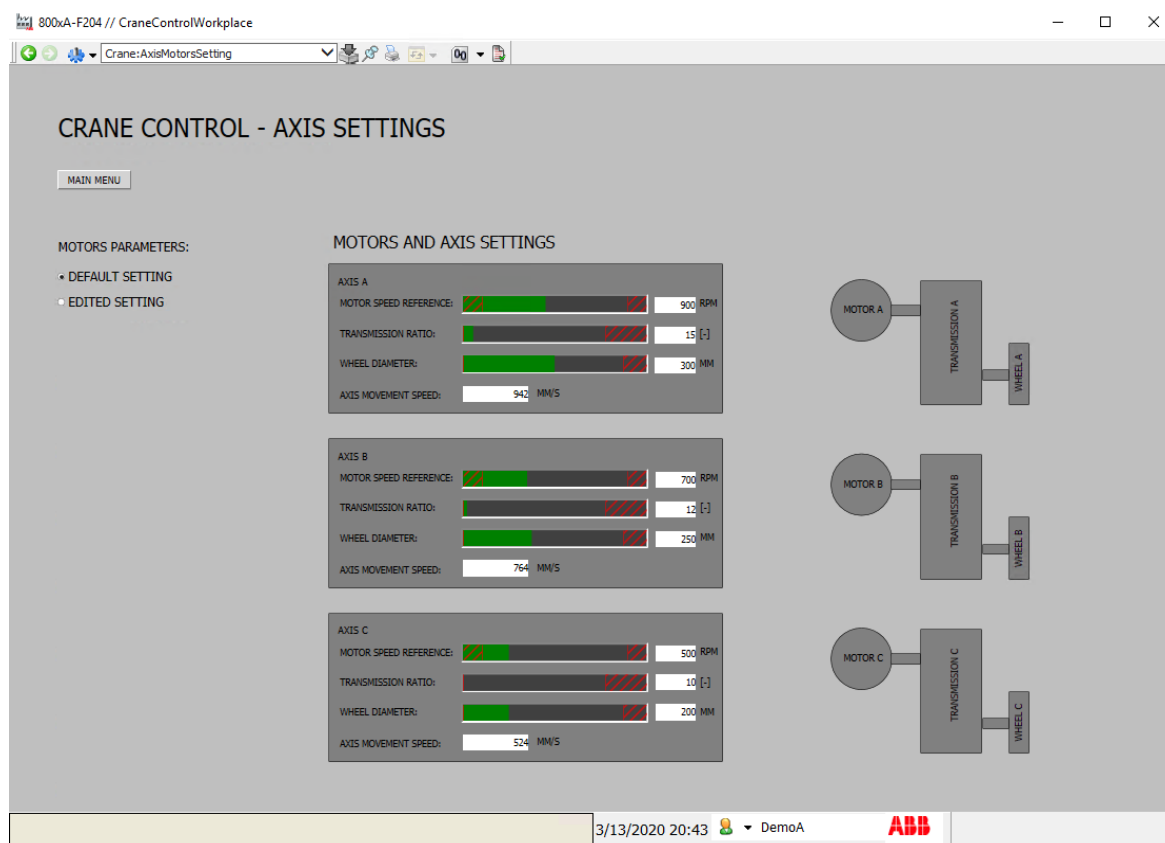
WORK POINT 2	DISPOSAL POINT	CRANE POINT
NAME: empty	NAME: empty	NAME: empty
TIME: 0	TIME: 0	TIME: 0
STATUS: PROGRESS	STATUS: PROGRESS	STATUS: PROGRESS
FINISHED: NO	FINISHED: NO	FINISHED: NO
PROGRESS: <div></div>	OK COUNT: 0 NOK COUNT: 0	ACTUAL POINT: LP TARGET POINT: LP
ABORT	RESET OK RESET NOK	LOAD

Obrázek 11.5 Detailní pohled na vybrané klíčové body

11.3 Nastavení os a motorů

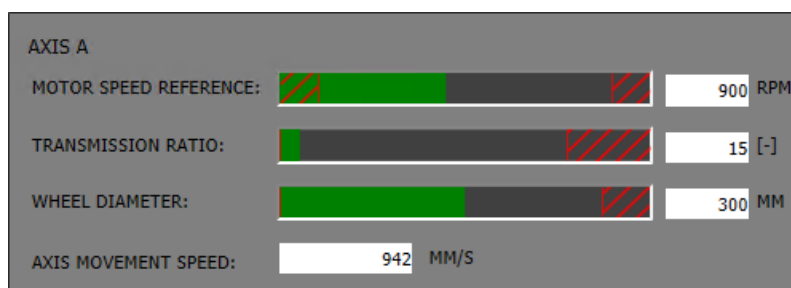
Jako další v pořadí byla zpracována obrazovka pro nastavení parametrů pro jednotlivé virtuální osy a motory. Na obrazovku bylo umístěno rozhraní pro volbu mezi výchozím a uživatelsky modifikovaným nastavením. Přepínání mezi módy bylo umožněno pouze v případě, že se demo úloha nenachází ani v automatickém, ani v ručním režimu a zároveň byl vypnut skutečný motor. Pokud byly výše zmíněné podmínky splněny, je možné mezi módy přepínat a případně upravovat nastavení os.

Výchozí nastavení vychází z požadavků na vyrovnanou rychlost pohybů v jednotlivých osách vycházející zejména z délky os. Při skutečné realizaci by bylo nutné rychlosti pohybů snížit úměrně k použité technologii (jeřáb nebo manipulátor) a zároveň by bylo nutné vzít v úvahu mnohé další faktory. Nastavení prezentovaná v demo úloze vycházejí z potřeby zkrácení času přesunů.



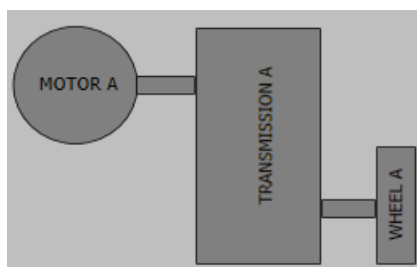
Obrázek 11.4 Obrazovka nastavení motorů a os - celkový pohled

Po označení volby uživatelských parametrů pro motory je možné nastavit jmenovitou rychlost otáčení motoru (skutečnou rychlost otáčení motoru při pohybu v dané ose), převodový poměr fiktivní převodovky a velikost pojezdového kola. Nastavení bylo realizována pomocí posuvníků, které jsou v horních a dolních oblastech doplněny o kritické oblasti, kterým by se měl operátor při nastavení raději vyhnout. Zároveň je vedle posuvníku explicitně uváděna nastavená hodnota daného parametru společně s jednotkami. Na základě zadaných údajů je operátorovy pod zadávacími posuvníky zobrazena výsledná pohybová rychlost v dané ose.



Obrázek 11.5 Nastavitelné parametry pro motor a osu A

Jako poslední element obrazovky byl pro jednotlivé osy vytvořen schematický náčrtek uspořádání motoru, převodovky a pohybového kola, ze kterého je možné pochopit logiku uspořádání pohonného a převodového mechanismu.



Obrázek 11.6 Náčrtek převodového ústrojí

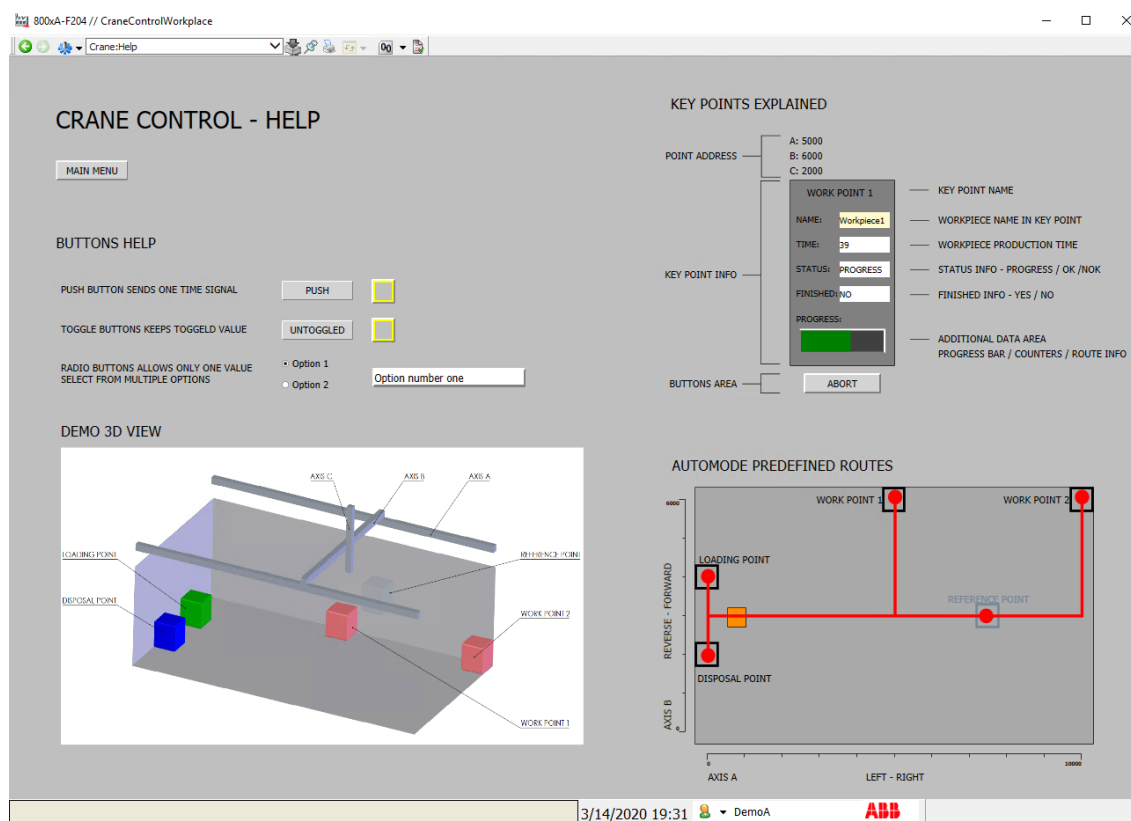
11.4 Obrazovka s nápovědou

Poslední operátorská obrazovka demo úlohy obsahuje nápovědu pro provoz aplikace a také vysvětlení jednotlivých grafických prvků. První kroky operátora by měli směřovat právě na obrazovku Help z důvodu nutnosti pochopení základních principů fungování aplikace.

Základní plán provozní místnosti je doplněn o vyznačené trasy automatického režimu. Červené čáry ukazují, kudy se pohybuje jeřáb, a červené tečky ukazují, kde je bezpečné se spustit z dopravní do nakládací výšky. Čáry a tečky navádějí operátora při ovládání jeřábu v manuálním módu. U skutečné realizace by mohlo být pro operátora nezbytné, aby s jeřábem manipuloval pouze ve vyznačeném prostoru a výškách z důvodu možné přítomnosti jiných objektů v cestě jeřábu mimo vyznačenou oblast.

Dále byl v nápovědě doplněn 3D pohled na výrobní halu s popisy jednotlivých os a výrobních bodů pro snadnější pochopení půdorysných mapek. V horní části obrazovky operátor nalezne několik demo tlačítek s vysvětlivkami jejich funkčnosti v demo úloze.

Poslední část obrazovky nápovědy patří vysvětlivkám pro informace zobrazované u jednotlivých klíčových bodů.



Obrázek 11.7 Přehled obsahu obrazovky s nápovědou

12 Dotykový panel a Program Panel Builder

Poslední částí demo úlohy bylo zprovoznění dotykového panelu PP874 od firmy ABB připojeného do společné sítě s PLC_1 a PLC_2. Panel je možné propojit napřímo s PLC nebo frekvenčním měničem. V případě demo úlohy bylo rozhodnuto o použití připojení pomocí Ethernetového portu do sítě TCP/UDP. Toto propojení umožňuje pracovat s daty z obou PLC. Na panelu je nutné provést základní konfiguraci a po připojení do sítě a k napájení je nutné nastavit panelu IP adresu. Do servisního menu panelu je možné vstoupit po restartu napájení, v okamžiku, kdy se na obrazovce objeví přesýpací hodiny, je nutné přitisknout prst na panel a čekat na další instrukce. V servisním menu je poté kromě IP adresy možné nastavit například jas, citlivost a další parametry odvíjející se od prostředí nasazení panelu. Příprava panelu k provozu je celkově velmi snadná a intuitivní, v okamžiku nastavení IP adresy a připojení do sítě je hardware připraven pro nahrání aplikace. [36]



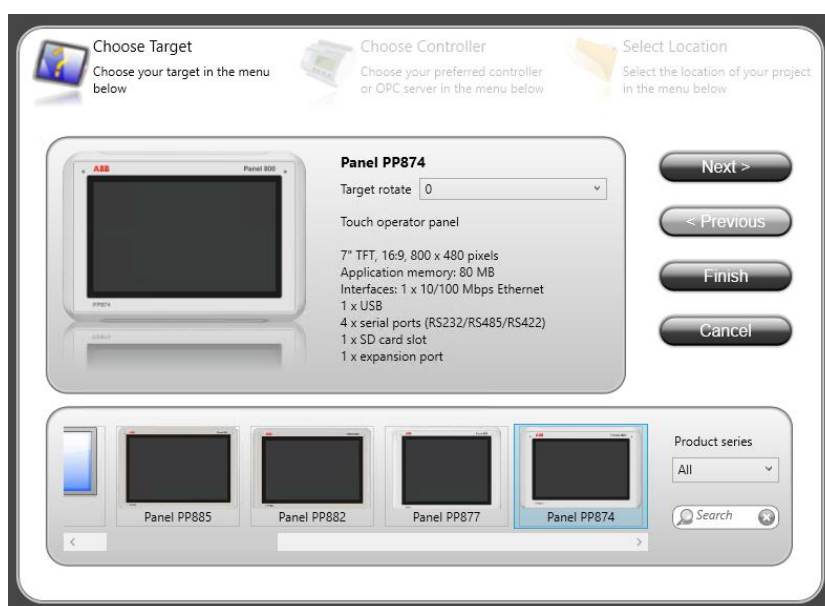
Obrázek 12.1 Úvodní obrazovka programu Panel Builder

K tvorbě aplikací pro dotykový panel slouží software Panel Builder 800. Jedná se o standalone software, který na rozdíl od programu Graphics Builder není přímo integrován do systému 800xA. Tato skutečnost v sobě skrývá určité výhody, ale také nevýhody. Díky samostatnosti programu nemusí koncový uživatel panelu provozovat distribuovaný systém a panel může být nasazen buď samostatně s PLC od ABB nebo s PLC jiného výrobce. Portfolio podporovaných výrobců PLC je poměrně široké a navíc je doplněno možností propojení s OPC serverem, což dělá z panelu hardware použitelný v jakémkoliv prostředí. Z důvodu důrazu na přenositelnosti a univerzálnosti ovšem utrpělo samotné vývojové prostředí, které je velmi zjednodušené. K dispozici je pouze omezené množství aktivních i pasivních objektů a vše je přizpůsobeno snadné ovladatelnosti a přehlednosti. Značným nedostatkem je zejména absence pokročilého editoru

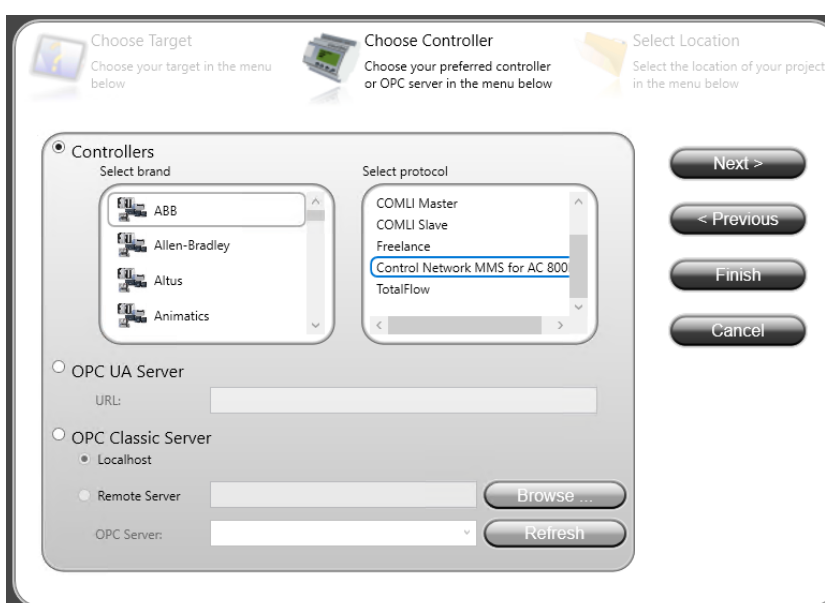
proměnných pro přípravu dat přicházejících z PLC, známého z programu Graphics Builder.
[37]

12.1 Vytvoření nového projektu a vývojového prostředí

Úvodní obrazovka programu nabízí volbu vytvoření nového projektu, nebo návrat k již vytvořenému projektu. Při tvorbě nového projektu je nutné vybrat správné typové označení panelu a definovat jeho orientaci. V dalším kroku je možné zvolit kontrolér, se kterým má panel spolupracovat. Obě nabídky jsou přístupné také později přímo z aplikace. V posledním kroku je ještě možné zadat jméno nového projektu a jeho umístění na lokálním disku počítače.[37]

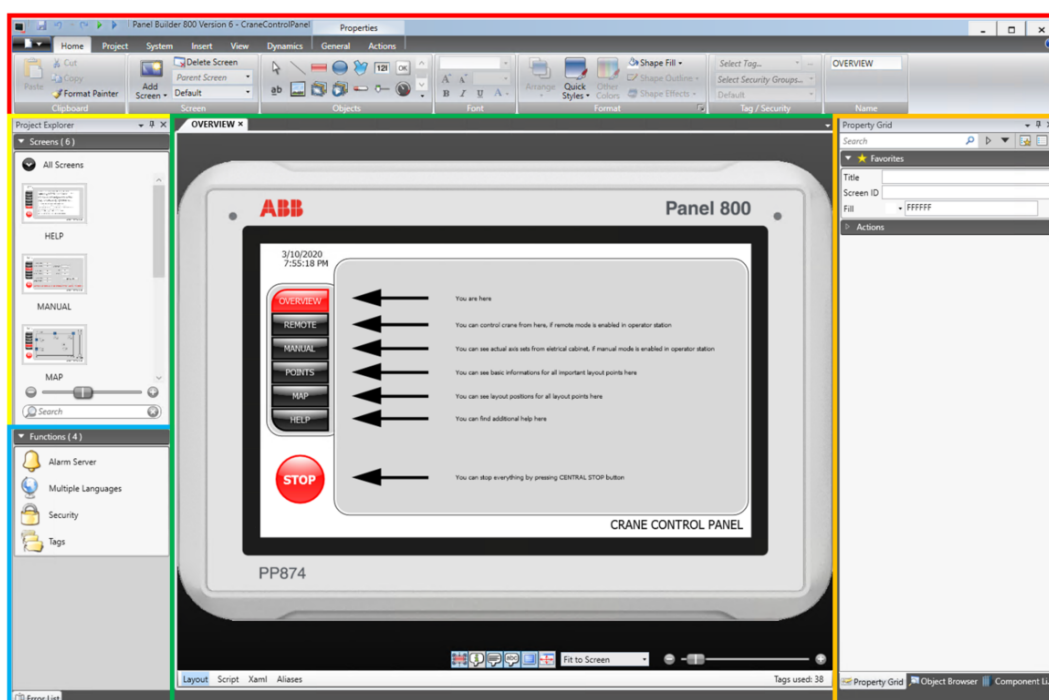


Obrázek 12.2 Konfigurace nového projektu



Obrázek 12.3 Výběr připojeného kontroléru

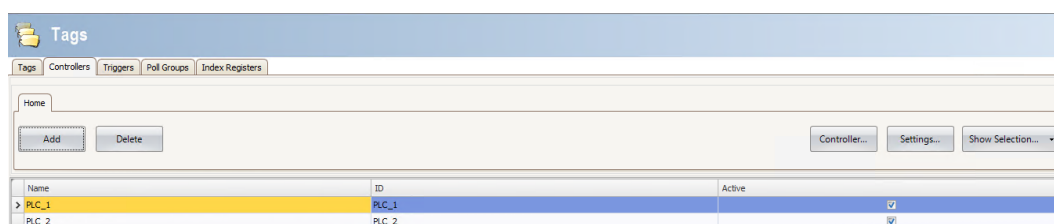
Po vytvoření nového projektu je uživatel přenesen do hlavního okna programu, odkud jsou přístupny všechny jeho základní funkcionality. V horní části obrazovky, na obrázku níže označené červeným rámečkem, je pás karet pro navigaci mezi vkládáním nových objektů, nastavením projektu, kompilací projektu a nahráním aplikace do panelu. Žlutě je označena oblast přehledu obrazovek, odkud je možná se přepínat mezi existujícími obrazovkami aplikace a také nastavit výchozí obrazovku. Modrým rámečkem je označena oblast výběru funkcionalit zahrnující nastavení alarmů, výběru jazyka, zabezpečení a také nastavení tagů. Zeleně označená oblast symbolizuje samotný panel včetně obrazovky, na kterou je možné vkládat jednotlivé objekty jako textová políčka, tvary, tlačítka, grafy a další. Poslední oranžově označená oblast slouží pro přístup k podrobnostem jednotlivých objektů vložených na obrazovku a jejich detailnímu nastavení. [37]



Obrázek 12.4 Program Panel Builder

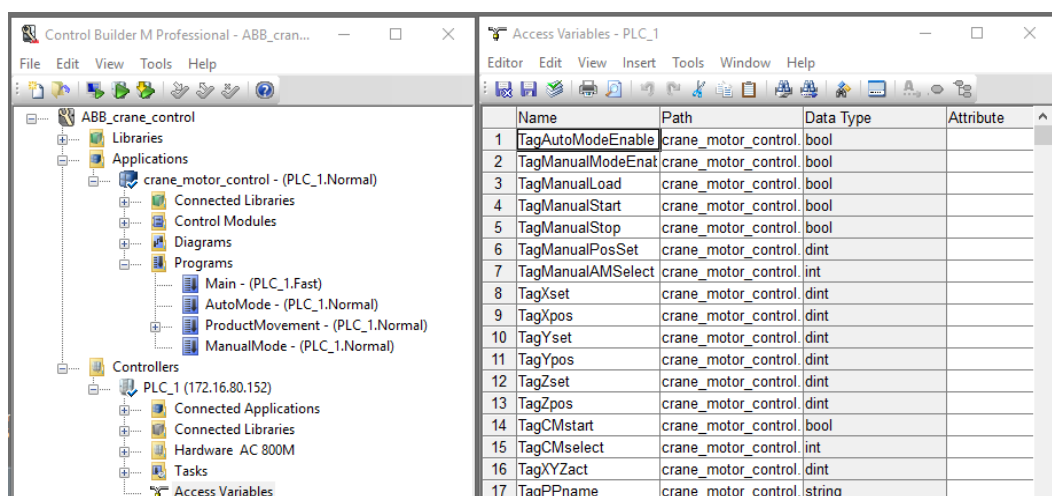
12.2 Propojení proměnných z PLC a v programu Panel Builder tagů

Pro komunikaci mezi aplikací a hardwarem připojeným na stejnou síť využívá program takzvaných Tagů. Tag představuje proměnnou, která může být v jednom ze základních datových typů a je napřímo propojena s proměnnými v PLC. Kromě popsanych základních předpokladů navíc disponuje možností nastavit k danému tagu přístupová práva a také je možné kontrolovat výměnu dat, aby nedocházelo k přetěžování komunikačních kanálů. Aby bylo možné tagy importovat, je nejprve nutné dokončit konfiguraci kontrolérů v záložce *Controllers*, jak je vidět na obrázku níže. Kontroléry by měli mít nastaveny správnou IP adresu, což je možné učinit z nabídky *Settings*. Do tabulky kontrolérů je možné kromě PLC nakonfigurovat také OPC server. [37]



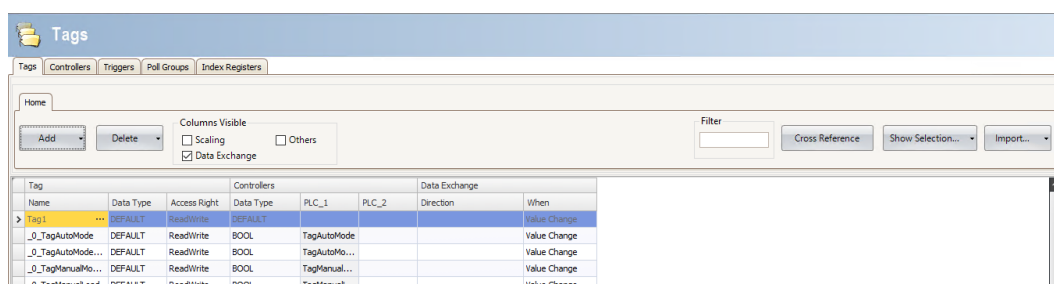
Obrázek 12.5 Tabulka kontrolérů v programu Panel Builder

Po dokončení nastavení kontrolérů je v případě použití PLC od ABB nutné v aplikaci Control Builder vytvořit Access Variables – přístupové proměnné umožňující propojení vnitřních proměnných v PLC s tagy aplikace Panel Builder. Na obrázku níže je možné vidět, že k přístupovým proměnným jednotlivých aplikací pro PLC je možné přistoupit pomocí položky Access Variables. Následně je nutné zadat nové jméno přístupové proměnné a také cestu k proměnné v PLC, na kterou se má přístupová proměnná navázat. Datový typ je doplněn automaticky a nové proměnné je možné nastavit atributy stejně jako při tvorbě kontrolní aplikace. [37]



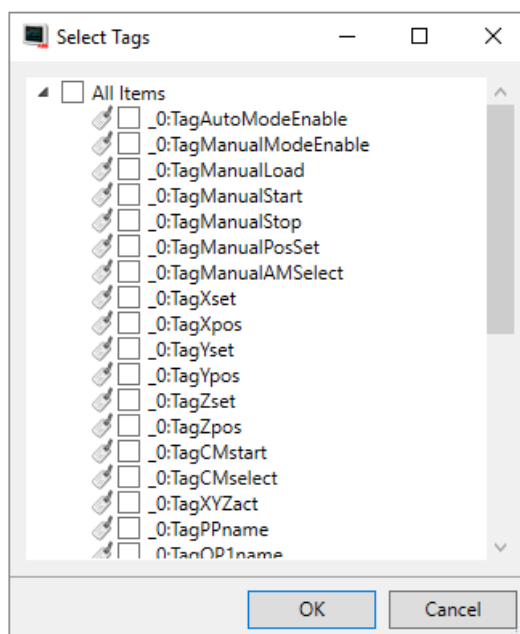
Obrázek 12.6 Přístupové proměnné v aplikaci ABB_crane_control

Jakmile je aplikace pro PLC zkompileována a aktualizována, je možné importovat přístupové proměnné jako tagy do programu Panel Builder. To je docíleno kliknutím na tlačítko *Import* a následném výběru kontroléru, ze kterého chceme importovat položky. Při importu jsou již viditelné přístupové proměnné z dané aplikace v PLC a je možné vybrat, které z nich budou nahrány. Dále je možné zadat, pod jakým jménem bude tag uložen. U většiny datových typů je program schopen při importu rozpoznat datový typ zdroje, ovšem u datových typů jako double, real, string a dalších je vhodné rozkliknout u daného tagu položku datového typu a přesvědčit se, že jsou správně nastaveny položky proměnné size. Implicitně je velikost nastavena na hodnotu jedna, což může vést k problémům se zobrazováním textových řetězců a čísel s desetinou čárkou.[37]



Tag			Controllers		Data Exchange	
Name	Data Type	Access Right	Data Type	PLC_1	PLC_2	When
Tag1	DEFAULT	ReadWrite	DEFAULT			Value Change
_0_TagAutoMode	DEFAULT	ReadWrite	BOOL	TagAutoMode		Value Change
_0_TagAutoMode...	DEFAULT	ReadWrite	BOOL	TagAutoMo...		Value Change
_0_TagManualMo...	DEFAULT	ReadWrite	BOOL	TagManual...		Value Change
_0_TagManualLoad	DEFAULT	ReadWrite	BOOL	TagManual...		Value Change

Obrázek 12.7 Tabulka Tagů v programu Panel Builder



Obrázek 12.8 Menu pro výběr importovaných tagů

12.3 Pasivní a aktivní prvky vizualizační obrazovky

Program Panel Builder je objektově orientovaný, avšak v základu disponuje pouze omezenou knihovnou objektů umožňující obsloužit základní funkcionality a požadavky na interakci. Nechybí tlačítka, textová pole, posuvníky, různé ukazatele a manometry, možnost vložit obrázek či tvar. K dispozici je také zobrazení trendů nebo přístup do databází. Většina prvků je ve výchozím tvaru svou velikostí a nastavením barev i kontrastů připravena pro dotykové panely. V případě potřeby je možné doinstalovat další objekty podobně jako při práci v program Graphics Builder.[37]

Tvorba obrazovky začíná výběrem možnosti nová obrazovka z pásu karet. K dispozici je několik výchozích šablon. Uživatel může vytvářet nové šablony. Pro efektivnost práce je vhodné nejprve vytvořit jednu obrazovku a upravit ji dle požadavků na vzhled i přítomnost funkčních prvků a následně ji uložit jako šablonu, ze které bude vycházeno při tvorbě dalších obrazovek. [37]



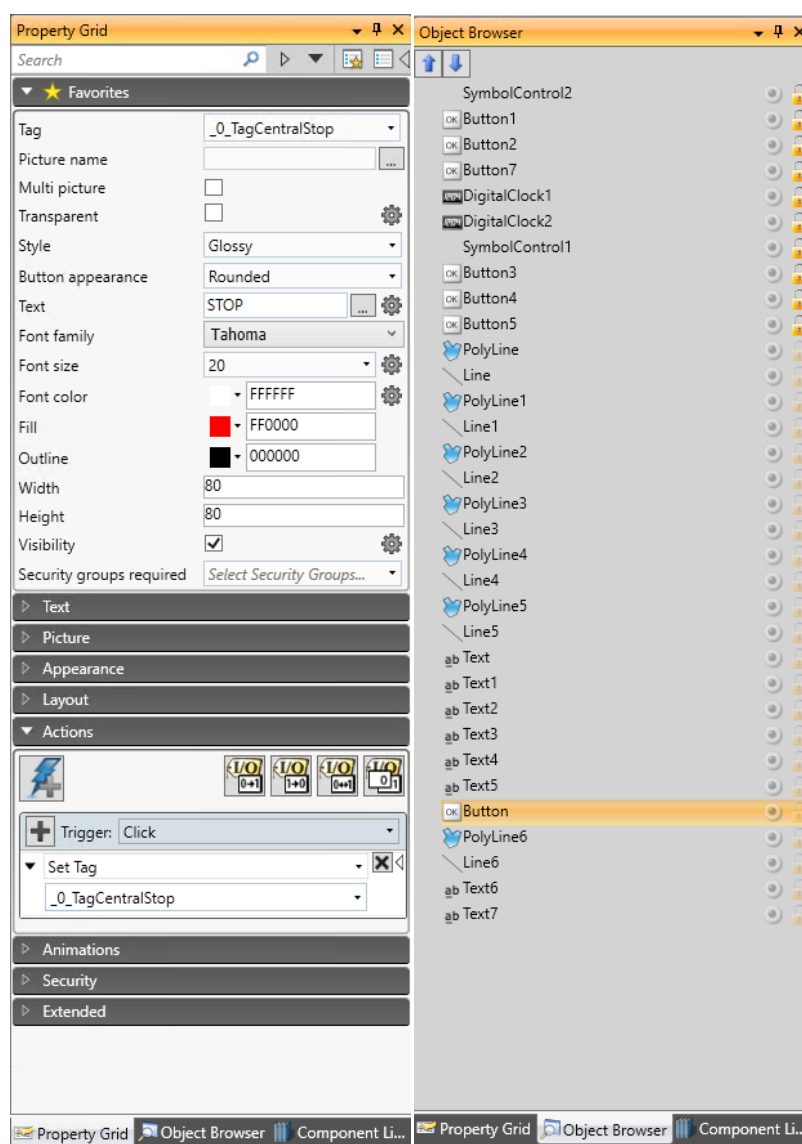
Obrázek 12.9 Seznam základních objektů

Po vytvoření obrazovky a umístění požadovaného objektu z nabídky na pásu karet na aktivní oblast obrazovky je možné tento objekt začít upravovat. Všechna nastavení jsou prováděna pomocí položky vlastností na pravém panelu obrazovky. V zásadě je možné měnit vzhledové položky jako velikost, umístění, velikost textu, rámeček, barvu nebo zarovnání textu. V nabídce oblíbené je možné také připojit výchozí tag, pokud se jedná o objekt určený například k zobrazování analogové hodnoty, bude se zobrazovat právě hodnota přiřazeného tagu. Velmi důležitou položkou vlastností je pak záložka Akce, která určuje chování objektů určených pro interakci s operátorem. V nabídce je velké množství akcí, které se po kliknutí mohou udát. Nejvyužívanějšími akcemi jsou zejména binární operace a obrazovkové operace. Binární operace slouží k nastavení hodnoty proměnné,

vynulování proměnné nebo k přepínání mezi true a false. Obrazkové operace slouží k přepínání mezi obrazkami. [37]

V případě, že je aplikace nakonfigurována pro různé uživatele s různým stupněm oprávnění, je možné nastavit také položku bezpečnosti obsluhující oprávnění jednotlivých uživatelů k přístupu k objektu. Dále je objektům možné nastavit různé vizuální efekty jako blikání, pohyb po obrazovce nebo proměnnou velikost.[37]

Při práci s větším počtem objektů na jedné obrazovce je nápomocná záložka Object Browser. Do okna jsou vypsány všechny objekty na obrazovce a je možné je pomocí dvou tlačítek buďto skrýt, nebo zamknout. Zamknutí objektu znepřístupní možnost editace a zároveň nedovolí náhodné posunutí na obrazovce. Při vytváření šablony obrazovky je vhodné objekty zamknout, aby nedošlo například k jejich náhodnému posunutí a s tím související nejednotě vizuálu jednotlivých obrazovek v aplikaci.[37]



Obrázek 12.10 Položky panelů Property Grid a Object Browser

12.4 Nahrání aplikace do panelu

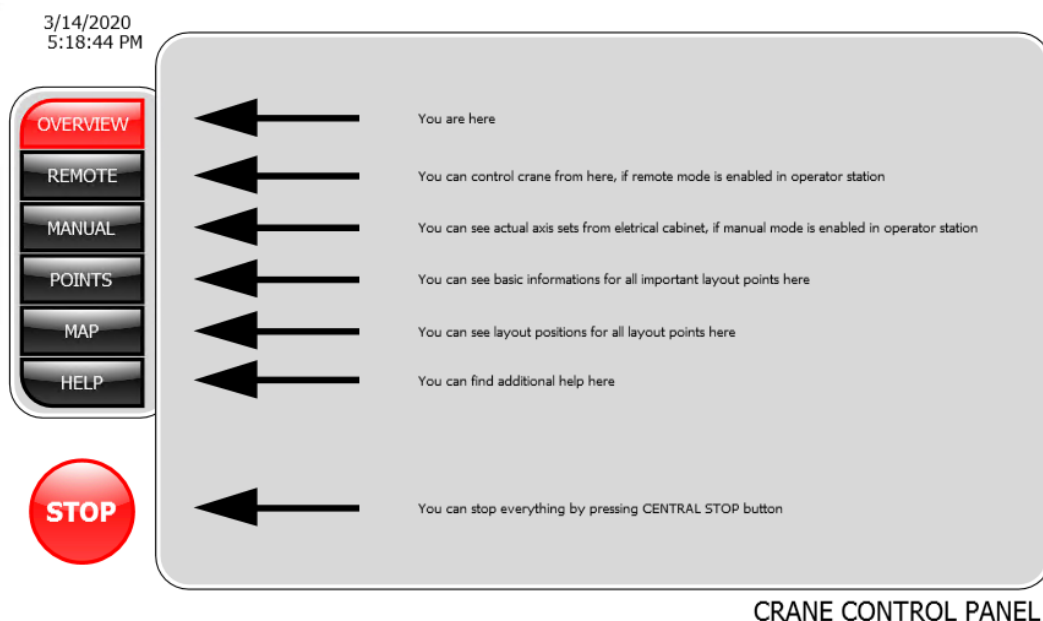
Po dokončení tvorby aplikace je nutné projekt zkompilevat a odstranit všechny případné chyby vypsané ve spodní konzole aplikace. Pokud je aplikace úspěšně a bez chyb zkompileována, může být přikročeno k nahrání do dotykového panelu výběrem možnosti Download z pásu karet. Následně se uživateli zobrazí seznam všech zařízení připojených na stejnou síť, do kterých je možné nahrávat. Seznam obsahuje informaci o IP adrese a typu zařízení, stejně jako název aktuálně používaného projektu. Jak je patrné z obrázku níže, uživatel může dle IP adresy specifikovat cílové zařízení a ověřit jeho připravenost na přenos nových souborů. Dále je možné zvolit volbu přenesení celého zdrojového kódu do zařízení a případně také na paměťovou kartu panelu. Tyto volby jsou vhodné zejména ve chvíli, kdy chce programátor zajistit zpětný přístup ke zdrojovým datům aplikace v panelu, které si může někdo jiný načíst přímo z konkrétního zařízení a není odkázán na další zdroje jako centrální knihovny projektů atd. Po provedení všech nastavení je možné stisknout tlačítko Download a projekt je následně přenesen a spuštěn na vybraném panelu.[37]



Obrázek 12.11 Nahrání aplikace do dotykového panelu

13 Vizualizační obrazovky pro dotykový panel PP874

Pro demo úlohu byla použita jedna ze základních šablon programu, která byla následně modifikována a doplněna. Na levé straně panelu se nachází nabídka jednotlivých obrazovek panelu se zvýrazněním aktuální obrazovky. Nad navigačním menu se nachází ukazatel data a času. Pod hlavním menu je pak k dispozici tlačítko STOP, které bylo funkčně namapováno na proměnnou CentralStop v řídicí aplikaci v PLC. Tlačítko STOP je stejně jako menu a ukazatel času dostupné ze všech nabídnutých obrazovek. Tlačítko slouží zejména jako nouzový prvek pro zastavení pohybu motoru i všech dalších procesů v případě nestandardního chování aplikace nebo hardwaru.

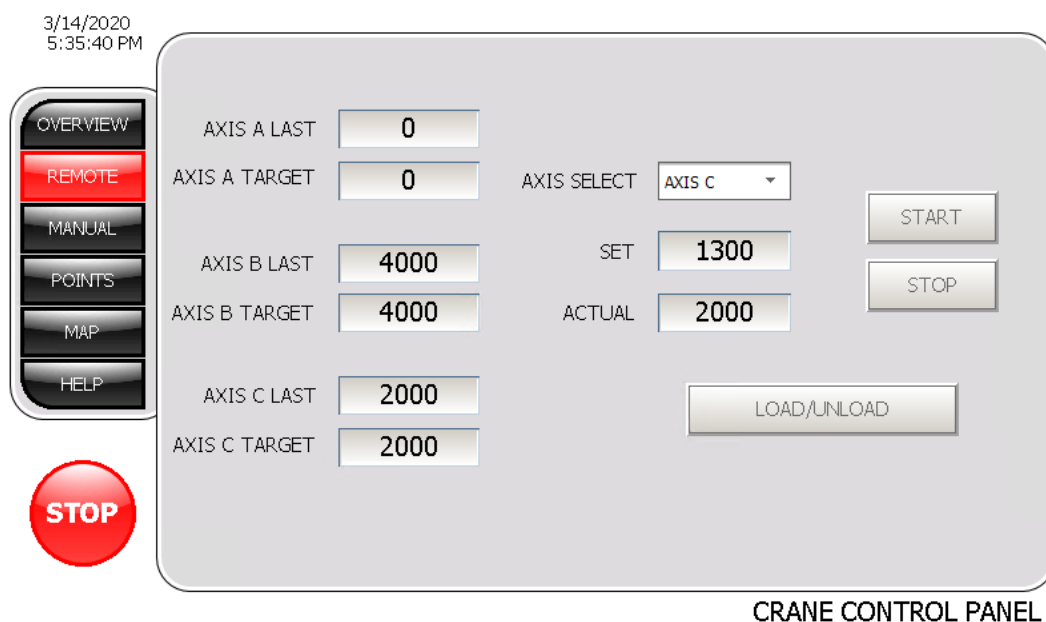


Obrázek 13.1 Úvodní obrazovka dotykového panelu

Úvodní obrazovka aplikace pro dotykový panel obsahuje krátkou nápovědu k jednotlivým položkám hlavního menu a slouží k navedení operátora na další obrazovky. Pro operátora jsou důležité zejména obrazovky REMOTE a MANUAL odpovídající již známému provozu. Přímou z panelu není z bezpečnostních důvodů možné volit mezi režimy provozu demo úlohy. Volbu režimu bylo umožněno provádět pouze z operátorské obrazovky pracovní stanice. Umožněno nebylo ani zadávání nových výrobků.

13.1 Obrazovka Remote

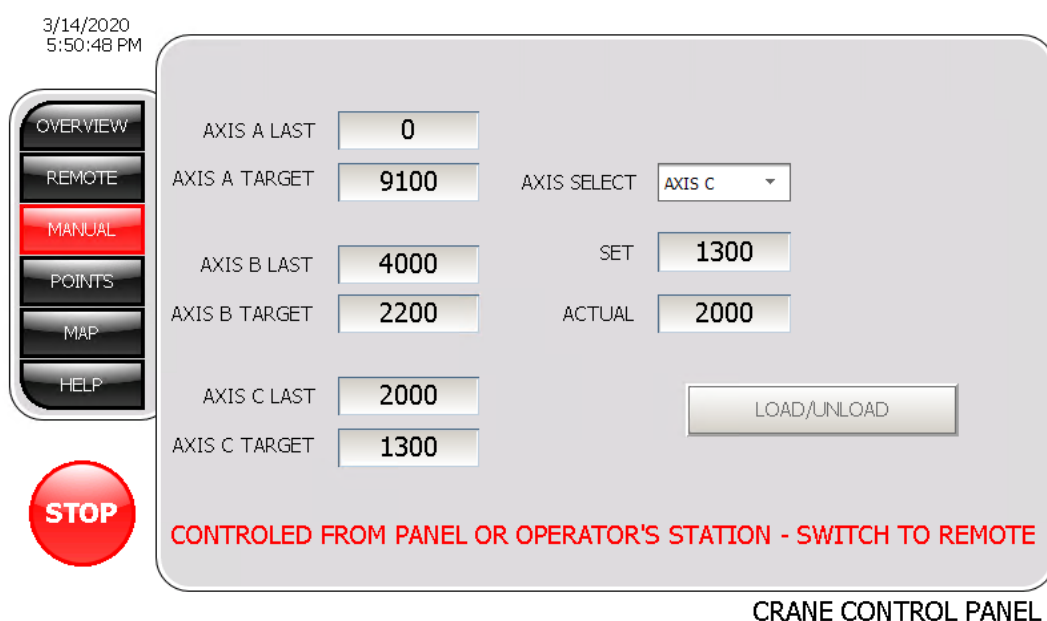
Druhá obrazovka aplikace s názvem REMOTE se vztahuje k módu vzdáleného ovládání, který je možné spustit z operátorského pracoviště na stolním počítači. Pokud je vzdálený režim aktivován, je operátorovi umožněno vybrat osu, nastavit požadovanou hodnotu a zastavit pohyb jeřábu. Pokud se nachází v jednom z klíčových bodů, může provést vykládku či nakládku výrobku pomocí příslušného tlačítka. V zásadě se jedná o přenesení ovládacích prvků vzdáleného režimu na dotykový panel a je jedno, zda zadávání požadovaných hodnot a spouštění pohybů probíhá z operátorského pracoviště nebo z dotykového panelu. Pokud mód není aktivován, ovládací prvky jsou vypnuté. Stále aktualizované však zůstávají zobrazovací políčka udávající současnou adresu jeřábu a také cílovou hodnotu pro jednotlivé osy. Obrazovka REMOTE je tak základní obrazovkou, na kterou je vhodné přepnout panel při práci z operátorské stanice, aby obsluha stále viděla aktuální informace o pohybu jeřábu.



Obrázek 13.2 Obrazovka Remote

13.2 Obrazovka Manual

Další neméně podstatná obrazovka s názvem MANUAL se příliš neliší od obrazovky REMOTE, přesto přináší některé další informace. Jak je patrné ze srovnání snímků obrazovky, na rozdíl od režimu REMOTE nezobrazuje požadované hodnoty na základě informací ze systému nebo ručního zadání, ale ukazuje hodnoty vyplývající z natočení potenciometrů na elektrickém rozvaděči. Obrazovka slouží zejména pro zobrazení nastavených hodnot operátorovi, který pak může také za pomoci potenciometrů přesně nastavit požadovanou hodnotu a pomocí tlačítka na rozvaděči aktivovat pohyb jeřábu. Na obrazovce zůstaly také ukazatele aktuálně vybrané osy, ze které je možné zjistit, ve které ose probíhá pohyb. Zůstalo také tlačítko pro překládku výrobků, které je stejně jako na obrazovce REMOTE aktivní pouze v okamžiku, že se jeřáb nachází v jednom z klíčových bodů. Červený text ve spodní části obrazovky v aplikaci problikává a upozorňuje operátora, že panel je v současnosti provozován v režimu REMOTE a je vhodné přepnout panel na příslušnou obrazovku pro zobrazení aktuálních hodnot. Obrazovka REMOTE pak obsahuje stejné upozornění aktivované v případě, že se aplikace nachází v manuálním módu.



Obrázek 13.3 Obrazovka Manual

13.3 Obrazovky Points, Map a Help

Pro operátora obsluhujícího demo úlohu od dotykového panelu je důležité znát obsah jednotlivých klíčových bodů. Za tímto účelem byla zařazena obrazovka se zjednodušenými informacemi o stavu jednotlivých klíčových bodů. Pro přípravný bod, nakládací bod a body řady jsou zobrazeny pouze o informace, jestli se v nich nachází nějaký výrobek. U bodu jeřábu a pracovních bodů jsou zobrazovány navíc informace o stavu dokončení výrobku. Pokud svítí nula u položky OK/NOK i u položky FINISHED, znamená to, že výrobek zatím nebyl dokončen. Pokud u položky FINISHED svítí hodnota jedna, znamená to, že výrobní operace byla dokončena a následně je možné z položky OK/NOK vyčíst, zda výroba proběhla úspěšně, nebo byla přerušena a výrobek je vadný. Jako poslední jsou zobrazovány informace o vykládkovém bodu společně s počítadlem dobrých a vadných výrobků.

3/14/2020
5:56:11 PM

OVERVIEW

REMOTE

MANUAL

POINTS

MAP

HELP

PREPARATION POINT: empty

CRANE POINT: empty

QUEUE POINT 1: empty

CRANE OK/NOK STATUS: 0

QUEUE POINT 2: empty

CRANE FINISHED STATUS: 0

LOADING POINT: empty

WORK POINT 1: empty

WORK POINT 2: empty

WP1 OK/NOK STATUS: 0

WP2 OK/NOK STATUS: 0

WP1 FINISHED STATUS: 0

WP2 FINISHED STATUS: 0

DISPOSAL POINT: empty

OK COUNT: 0

NOK COUNT: 0

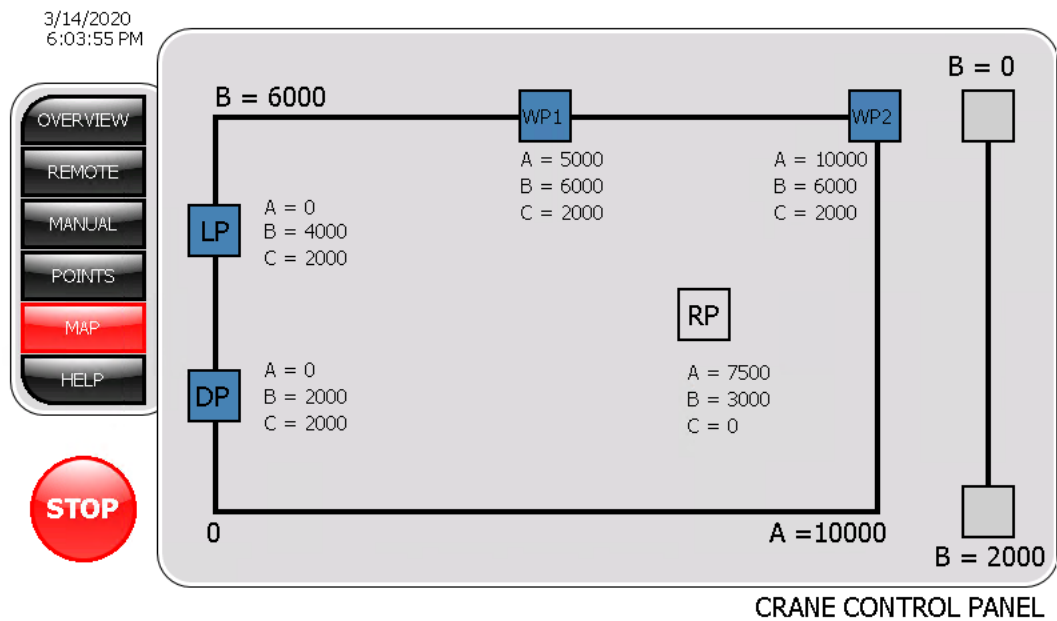
STOP

CRANE CONTROL PANEL

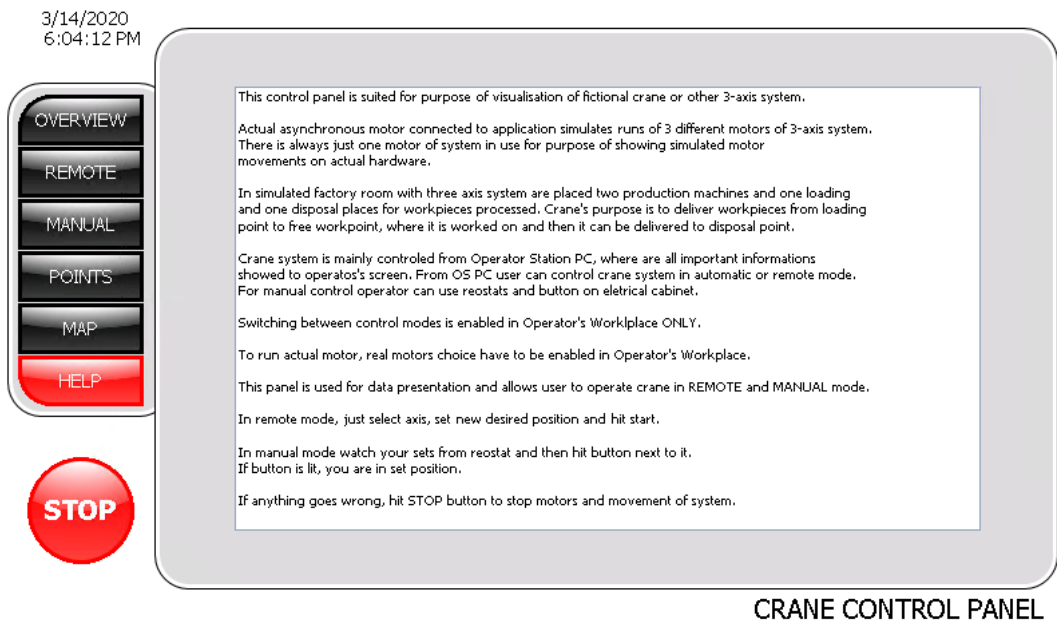
Obrázek 13.4 Obrazovka s informacemi o klíčových bodech

Obrazovka MAP je podpůrná obrazovka bez aktivních prvků. Je na ní znázorněna mapa výrobních prostor společně s adresami jednotlivých klíčových bodů. Zároveň je na obrázku vpravo naznačeno, že nulová hodnota osy C sloužící k vyzvednutí výrobku je pod stropem místnosti a jedná se o dopravní výšku.

Poslední obrazovka s názvem HELP obsahuje podrobnější nápovědu k ovládání panelu a jsou na ni zapsány informace o chování panelu v jednotlivých módech, stejně jako pravidla pro přepínání obrazovek.



Obrázek 13.5 Obrazovka Map



Obrázek 13.6 Obrazovka s nápovědou

14 Bezpečnost a uživatelská příručka

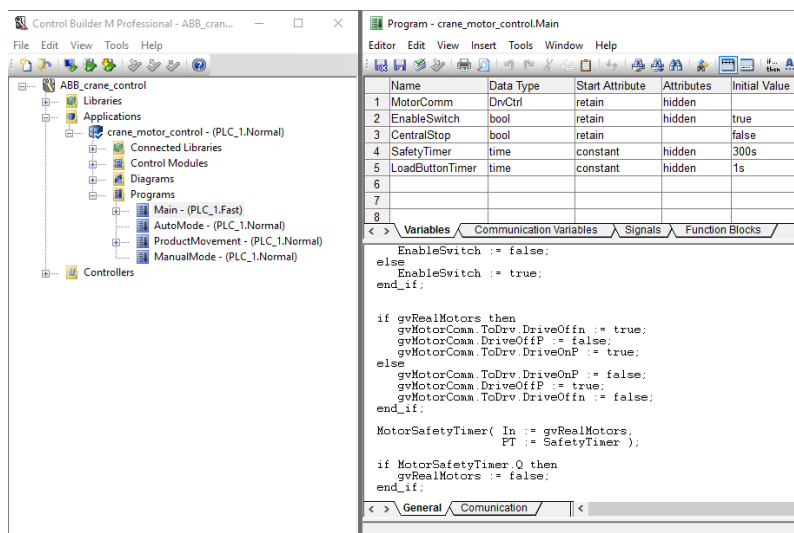
Následující kapitola shrnuje základní bezpečnostní opatření k provozu demo úlohy, věnuje se představení uživatelské příručky a obsahuje popis nové infografiky umístěné na čelní stranu elektrického rozvaděče pro provoz demo úlohy v manuálním režimu.

14.1 Bezpečnostní pravidla demo úlohy

Pro budoucí bezpečný provoz demo úlohy je nutné nejprve nastudovat uživatelskou příručku dodanou k pracovišti. Přestože jednotlivé ovládací prvky mají dostatečný stupeň krytí pro neproškolenou obsluhu, stále je nutné dbát zvýšené opatrnosti při kontaktu s PLC umístěným mimo rozvaděč. Rozvaděč je zakázáno otevírat bez dozoru pověřené osoby. Dále je zakázáno jakékoliv součásti demo úlohy odpojovat. Kompletní souhrn pokynů pro provoz obsahuje uživatelská příručka.

Dalším ze základních prvků přispívajících k větší bezpečnosti pracoviště je skutečnost, že byl provozu reálných motorů nastaven časový limit běhu na 5 minut. Po pěti minutách dojde k automatickému vypnutí skutečného motoru. Toto opatření není možné deaktivovat a slouží zejména k ochraně motoru proti přehřátí a zároveň napomáhá zamezit zvýšené spotřebě elektrické energie ve chvíli, kdy se demo úloze nikdo nevěnuje, případně ji omylem nechal běžet.

Mezi další bezpečnostní opatření patří přítomnost STOP tlačítka na všech obrazovkách dotykového panelu, pomocí kterého je možné přerušit provoz demo úlohy a následně ji restartovat. Další spuštění úlohy bylo umožněno pouze z prostředí operátorského pracoviště, ke kterému bude na rozdíl od dotykového panelu mít přístup pouze proškolená obsluha.



Obrázek 14.1 Doplnění bezpečnostního časovače do programu pro PLC

14.2 Uživatelská příručka

Příručka primárně slouží k rychlému zaškolení obsluhy demo úlohy a shrnuje všechny její funkcionality. Kromě jiného jsou v ní uvedeny bezpečnostní pravidla provozu demo úlohy. V poslední řadě pak obsahuje krátké vysvětlení principu distribuovaného systému řízení. Uživatelská příručka je dostupná v příloze D.

Úvod příručky obsahuje bezpečnostní pokyny pro provoz demo úlohy, kterými je povinný se uživatel řídit po celou dobu provozu. Následně jsou v příručce uvedeny přihlašovací údaje společně s popisem architektury demo úlohy. Důležité jsou pokyny pro spuštění distribuovaného systému řízení, včetně návodu ke spuštění operátorského pracoviště na virtuální pracovní stanici. Podobně byla popsána také sekvence pro vypnutí celého systému, která musí být bezpodmínečně dodržována.

Dále je v příručce popsán princip demo úlohy a operátorovi je vysvětlena funkce jednotlivých vizualizačních obrazovek operátorského pracoviště a dotykového panelu. K zajištění rychlého přehledu o ovládacích prvcích jednotlivých obrazovek byla do příručky přidána kapitola popisující funkcionality všech dostupných tlačítek.

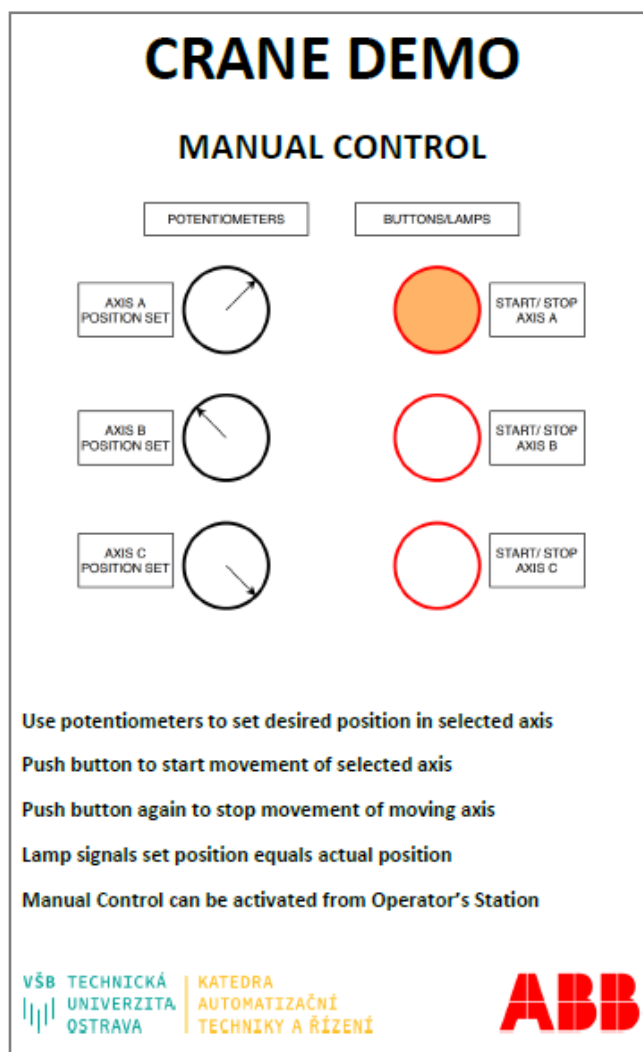


Obrázek 14.2 Titulní strana uživatelské příručky demo úlohy

14.3 Manuální režim

Pro potřeby manuálního režimu byla vytvořena nová infografika umístěná na elektrický rozvaděč. Infografika obsahuje informace o způsobu ovládání jeřábu skrze tlačítka a potenciometry a zároveň podává vysvětlení pro světelnou signalizaci. Je dostupná v příloze E diplomové práce.

Při otáčení potenciometry se mění hodnota požadované pozice pro jednotlivé osy. Pokud je nastavená hodnota na potenciometru shodná s aktuální hodnotou osy v systému, rozsvítí se lampička. Odstartovat pohyb ve vybrané ose je možné stlačením příslušného tlačítka. Stejným pokynem je možné pohyb v ose také zastavit. Aktuálně nastavené hodnoty může operátor sledovat na dotykovém panelu, odkud může příslušným tlačítkem také provést přesun výrobku mezi klíčovými body.



Obrázek 14.3 Infografika manuálního módu

15 Závěr

Prvním úkolem diplomové práce bylo získání základního vhledu do problematiky distribuovaného systému řízení. V prvních kapitolách byly vytyčeny důležité pojmy související s DCS a SCADA systémy. Dále bylo popsáno několik v současnosti používaných systémů v průmyslové praxi. Následující kapitoly se věnovali revizi stávajícího stavu demo úlohy a její hardwarové a softwarové vybavenosti. Byl popsán asynchronní motor, PLC a další použité komponenty. Původní program pro řízení motoru byl úspěšně otestován.

Na výchozí konfiguraci navázala úprava pracoviště přidáním dotykového ovládacího panelu a instalace pokročilého řídicího systému 800xA od firmy ABB. K důležitým bodům patřil popis principu virtualizace, díky které je možné provozovat na jednom lokálním počítači hned několik virtuálních strojů v přívětivém a přehledném uživatelském prostředí. Výše uvedené principy byly využity při realizaci demo úlohy. Díky práci s virtuálními stroji vzrostla životnost úlohy, která již není závislá na místním hardwaru, ale stala se dobře přenositelnou v případě nutnosti přechodu na nový operační systém, nebo výpočetní hardware.

V následující části byla vypracována funkční specifikace popisující novou demo úlohu a byly definovány požadavky na její funkčnost ve třech módech. Demo úloha představuje výrobní halu s jeřábem nebo tříosým manipulátorem, který se stará o rozvoz výrobků mezi klíčovými body určenými pro nakládání výrobků, výrobu a odkládání. Výše uvedené operace je možné provádět v automatickém, vzdáleném a ručním režimu. Kromě jiného byly v této části také předběžně navrženy obrazovky operátorské stanice a operátorského panelu určené k ovládání demo úlohy.

Funkční specifikaci následovalo vypracování vývojového diagramu popisujícího funkci aplikace a zpracování pravdivostní tabulky automatického režimu. Následovala příprava virtuálních počítačů a distribuovaného systému řízení ABB 800xA ve spolupráci s odborníky z firmy ABB. V další části projektu byly nakonfigurovány vybraná PLC, byl aktualizován jejich firmware a kontroléry byly připraveny pro naprogramování nové demo úlohy.

Díky implementaci virtualizace bylo možné na relativně malém počtu fyzických zařízení efektivně prezentovat principy distribuovaného systému řízení včetně přihlašování s různým oprávněním a různými rolemi.

Nová demo úloha byla sestavena za pomoci přenesení knihovny původního Control Modulu určeného k řízení motoru do systému 800xA a také implementací nového funkčního diagramu a příslušných programů pro řízení chodu aplikace a jejích jednotlivých

režimů. Následně byly vytvořeny všechny části programu, které byly nejprve simulačně otestovány a následně nahrány do hardwaru.

Po naprogramování nové aplikace byla ověřena funkčnost modelu pomocí otestování kombinace všech možných provozních stavů a na základě testování byly do kódu doplněny podmínky pro správný chod programu a přechodů mezi režimy aplikace.

Dalším významným blokem diplomové práce byla příprava vizualizací. Nejprve bylo v systému 800xA s ohledem na dostupné zobrazovací prostředky naprogramováno nové operátorské pracoviště určené pro řízení demo úlohy a následně bylo přistoupeno k tvorbě vizualizačních obrazovek. Pro aplikaci byly připraveny celkem čtyři obrazovky sloužící ke kompletní správě demo úlohy. Jako poslední byly vytvořeny vizualizační obrazovky pro dotykový panel. Přidání dotykového panelu přispělo k lepší orientaci operátora v probíhajících procesech demo úlohy.

Po dokončení všech dílčích prací bylo přistoupeno ke komplexnímu testování. Na základě poznatků z dlouhodobého provozu byly provedeny úpravy operátorských obrazovek vedoucí k zvýšení uživatelské přívětivosti aplikace. Značná pozornost byla věnována bezpečnému provozu demo úlohy.

Z důvodu potřeby rychlého zaškolení obsluhy demo úlohy distribuovaného systému řízení byla vytvořena uživatelská příručka obsahující bezpečnostní pokyny, popis architektury systému, souhrn použitých komponent a instrukce ke spuštění a provozu. Příručka obsahuje popis principu demo úlohy a také přehled všech operátorských obrazovek jak pro pracovní stanici, tak pro dotykový panel. Důležitý je také přehled všech dostupných tlačítek pro jednotlivé obrazovky spolu s vysvětlením jejich funkce. Pro ovládací prvky umístěné na elektrickém rozvaděči byla vytvořena nová infografika.

Demo úloha byla úspěšně dokončena a nasazena. Podařilo se zahrnout všechny požadavky vycházející z funkční specifikace. V úloze byly využity dvě PLC, jedno pro řízení frekvenčního měniče a asynchronního motoru, druhé pro obsluhu tlačítek s podsvícením a potenciometrů. Na dvou osobních počítačích běží v hypervizech celkem šest virtuálních počítačů s distribuovaným systémem řízení ABB 800xA. Nedílnou součástí úlohy je také dotykový panel.

Z uživatelského hlediska se při komplexním testování ukázalo jako nejpřívětivější obsluhovat demo úlohu z operátorského pracoviště na osobním počítači. Operátor má v tomto případě největší přehled o důležitých částech demo úlohy a úkony určené k vykonávání operátorem mohou být prováděny z jediné obrazovky obsahující všechny klíčové informace a ovládací prvky. Při využívání dotykového panelu dochází v porovnání

s operátorským pracovištěm ke ztrátě přehlednosti. Z důvodu malé velikosti panelu je operátor nucen často přepínat mezi obrazovkami a bylo zjištěno, že práce s panelem je z dlouhodobého hlediska výrobní efektivity nevhodná. Panel by měl být primárně určen pro zvýšení přehlednosti o procesu a k výjimečným zásahům operátora. Ani ovládání skrze tlačítka a potenciometry nedosahuje komfortu operátorského pracoviště. Režim má pomalou odezvu a z důvodu omezeného počtu nastavovacích prvků je možné úlohu provozovat pouze společně s dotykovým panelem zobrazujícím informace o nastavených hodnotách.

Diplomová práce byla zaměřena na supervizní systémy a úkolem demo úlohy bylo ukázat možnosti propojení hardwaru a softwaru v distribuovaném systému řízení. Z důvodu výše popsaného směřování práce bylo rozhodnuto o zavedení některých zjednodušení, která by mohla být dále rozvíjena. Z důvodu zpřístupnění aplikace pro pozdější úpravy a vylepšení byly jednotlivé moduly demo úlohy v řídicí aplikaci vzájemně odděleny. Z pohledu různých oborů věnujících se automatizační technice by bylo zajímavé rozpracovat algoritmus pro řízení otáček motoru v závislosti na provozních podmínkách demo úlohy, zejména na váze a tvaru přepravovaného výrobku. Z pohledu optimalizace se nabízí testování různých algoritmů pro výběr trasy manipulátoru v závislosti na stavu klíčových bodů. Za zmínku stojí také možnost propojení distribuovaného systému řízení s informačním systémem a zkompletování automatizační pyramidy.

V diplomové práci se podařilo splnit všechny body zadání. Nejprve byl charakterizován současný stav problematiky supervizních systémů a následně byla navržena nová demo úloha. Realizace navržené demo úlohy proběhla v distribuovaném systému řízení a byla úspěšně dokončena včetně všech požadovaných funkcionalit. Funkčnost demo úlohy byla úspěšně otestována. Pro potřeby výuky byla vytvořena uživatelská příručka. Demo úloha je díky důrazu na modularitu programového řešení připravena na další rozšíření.

Poděkování

Rád bych poděkoval vedoucí diplomové práce doc. Ing. Lence Landryové, CSc. za rady, připomínky a vstřícnost při konzultacích. Zároveň chci poděkovat rodině a přátelům za podporu v celém průběhu studia.

16 Seznam použité literatury

- [1] SHERIDAN, Thomas B. *Telerobotics, automation, and human supervisory control*. Cambridge, Mass.: MIT Press, 1992. ISBN 978-0-262-19316-0.
- [2] SCHOLTEN, Bianca. *The road to integration: a guide to applying the ISA-95 standard in manufacturing*. Research Triangle Park NC: ISA, 2007. ISBN isbn978-0-9792343-8-5.
- [3] What is SCADA?. In: *YouTube* [online]. RealPars, 2019 [cit. 2019-06-23]. Dostupné z: <https://www.youtube.com/watch?v=nIFM1q9QPJw>
- [4] System 800xA - distribuovaný řídicí systém (DCS). *ABB* [online]. Curych: ABB, 2019 [cit. 2019-06-24]. Dostupné z: <https://new.abb.com/control-systems/cs/800xa/800xa-dcs>
- [5] WHAT IS DCS?. *RealPars* [online]. RealPars, 2019 [cit. 2019-07-18]. Dostupné z: <https://realpars.com/dcs/>
- [6] PAPCUN, Peter, Erik KAJATI a Jiri KOZIOREK. Human Machine Interface in Concept of Industry 4.0. In: *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)* [online]. IEEE, 2018, s. 289-296 [cit. 2019-06-26]. DOI: 10.1109/DISA.2018.8490603. ISBN 978-1-5386-5102-5. Dostupné z: <https://ieeexplore.ieee.org/document/8490603/>
- [7] LYNGAAS, Sean. Nuclear Power Plants Have a 'Blind Spot' for Hackers. Here's How to Fix That. In: *Pulitzer Center* [online]. Washington, DC: Pulitzer Center on Crisis Reporting, 2018 [cit. 2019-06-28]. Dostupné z: <https://pulitzercenter.org/reporting/nuclear-power-plants-have-blind-spot-hackers-heres-how-fix>
- [8] Augmented Reality in Industry 4.0 – A Great Gimmick or Potential Industry Changer?. In: *Machines4u* [online]. Broadbeach: machines4u, 2017 [cit. 2019-06-26]. Dostupné z: https://db58mjtjr0n9n.cloudfront.net/wp-content/uploads/2017/05/17153016/REFLEKT-AR_recuperator_02_parts-759x500.jpg

- [9] PTAK, Richard, J. P. MORGENTHAL a Simon FORGE. *Manager's guide to distributed environments: from legacy to living systems*. New York: John Wiley, 1999. ISBN 0-471-19712-2.
- [10] LANDRYOVÁ, Lenka, Martin PAVELEK a Michal KONEČNÝ. *Návrh procesních systémů: programové systémy SCADA/MMI*. Vyd. 1. Ostrava: VŠB-Technická univerzita, 1996. ISBN 80-02-01100-7.
- [11] LANDRYOVÁ, Lenka. *Návody ke cvičení z předmětu Procesní systémy: učební text k předmětu "Procesní systémy"*. První. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2011. ISBN 978-80-248-2765-0.
- [12] KAPSA, Miroslav. *Grafické zobrazení dat z průmyslových aplikací* [online]. Ostrava, 2014 [cit. 2019-03-20]. Dostupné z: <http://hdl.handle.net/10084/104788>.
Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.
- [13] *Pantek* [online]. Hradec Králové: Pantek, 2019 [cit. 2019-07-19]. Dostupné z: <http://www.pantek.cz/index.php>
- [14] ŠKUTA, Jaromír. *Návrh distribuovaných řídicích systémů a jejich komunikační vazby: teze habilitační práce k habilitačnímu řízení v oboru Řízení strojů a procesů*. 1. vydání. Ostrava: VŠB - Technická univerzita Ostrava, 2016. ISBN isbn978-80-248-3968-4.
- [15] ABB Ability. *ABB* [online]. Zurich: ABB, 2019 [cit. 2019-07-21]. Dostupné z: <https://new.abb.com/abb-ability>
- [16] *ABB* [online]. Curych: ABB, 2019 [cit. 2019-07-11]. Dostupné z: <https://new.abb.com/cz>
- [17] KVÁČ, Jan. *SIMATIC PCS7 - Úvod do systému v8.0*. Praha, 2012. Dostupné také z: https://w5.siemens.com/web/cz/cz/corporate/portal/home/produkty_a_sluzby/IA-DT/tia_na_dosah/Documents/_1____SIMATIC_PCS_7_uvod.pdf
- [18] *InTouch* [online]. Hradec Králové: Pantek, 2019 [cit. 2019-07-21]. Dostupné z: <http://www.intouch.cz/index.htm>

- [19] Virtualization: A Complete Guide. In: *IBM* [online]. Armonk: IBM, 2019 [cit. 2019-06-26]. Dostupné z: https://www.ibm.com/cloud/learn/virtualization-a-complete-guide?mhq=virtualization&mhsrc=ibmsearch_a
- [20] KILIÁN, Karel. Více systémů na jednom počítači: Vybrali jsme nejlepší programy pro virtualizaci. In: *Živě.cz* [online]. Praha: CZECH NEWS CENTER a.s., 2019 [cit. 2019-06-26]. Dostupné z: <https://www.zive.cz/clanky/vice-systemu-na-jednom-pocitaci-vybrali-jsme-nejlepsi-programy-pro-virtualizaci/sc-3-a-195920/default.aspx#part=1>
- [21] Virtualizace. In: *Oldany Group* [online]. Praha: Oldany Group, 2019 [cit. 2020-03-19]. Dostupné z: https://lh3.googleusercontent.com/proxy/y2xi6ti6WAHCVlKitClrweGacAiCtbn_oh1YL2rLziFMjkbJeZ1zsr1amXLpfOiFIUYS1utrmMZD8bRJkmZwru8bks2LnzKmoPCTW2YaQtRCr1T
- [22] *Workstation 15 Is Here* [online]. In: . Palo Alto: VMware, 2019 [cit. 2019-06-26]. Dostupné z: <https://blogs.vmware.com/workstation/files/2018/09/WS-UI.png>
- [23] *Microsoft* [online]. Redmont: Microsoft, 2019 [cit. 2019-06-28]. Dostupné z: <https://www.microsoft.com/>
- [24] ZAVADIL, Jaromír. *Systém monitorování a řízení technologie* [online]. Ostrava, 2009 [cit. 2019-03-20]. Dostupné z: <http://hdl.handle.net/10084/71519>. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.
- [25] STAVĚLA, Robert. *Distribovaný systém řízení 800xA s kontrolerem AC500, OPC komunikací a vizualizací* [online]. Ostrava, 2018 [cit. 2019-03-20]. Dostupné z: <http://hdl.handle.net/10084/129708>. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.
- [26] Samostatné nástěnné frekvenční měniče ACS800-01. *ABB* [online]. Curych: ABB, 2019 [cit. 2019-07-11]. Dostupné z: <https://new.abb.com/drives/cs/nizkonapetove-frekvencni-menice/prumyslova-zarizeni/prumyslove-menice/samostatne-frekvencni-menice-ac800/acs800-01>
- [27] AC 800M Controller. *ABB* [online]. Curych: ABB, 2019 [cit. 2019-03-20]. Dostupné z: <https://new.abb.com/control-systems/essential-automation/compact-product-suite/essential-controller-suite/ac-800m-controller>

- [28] Compact Control Builder for AC 800M Controller. *ABB* [online]. Curych: ABB, 2019 [cit. 2019-07-16]. Dostupné z: <https://new.abb.com/control-systems/essential-automation/compact-product-suite/essential-controller-suite/compact-control-builder>
- [29] PP874. *ABB Compact Hardware Selector* [online]. Curych: ABB, 2019 [cit. 2019-07-17]. Dostupné z: <https://www.compacthardwareselector.com/product/pp874>
- [30] Overhead Crane. In: *Yale Forklifts* [online]. Quebec, CANADA: Yale Forklifts, 2019 [cit. 2019-10-30]. Dostupné z: https://www.yaleforklifts.com/files/download/image/600x0/FIXED_WIDTH/96a856e0-cb86-4637-bfff-94fc9594cb4c.png
- [31] GURÁŠ, Radek. *Model automatického parkovacího domu a jeho řízení* [online]. Ostrava, 2018 [cit. 2019-10-26]. Dostupné z: <http://hdl.handle.net/10084/130302>. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.
- [32] LIBERDA, Tomáš. *Řízení pojezdu mostového jeřábu pomocí frekvenčního měniče prostřednictvím programovatelného automatu Simatic S300* [online]. Ostrava, 2009 [cit. 2019-10-26]. Dostupné z: <http://hdl.handle.net/10084/74008>. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.
- [33] STEKLÝ, Jakub. *Model mostového jeřábu* [online]. Brno, 2016 [cit. 2019-10-26]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/94404>. Bakalářská práce. VUT Brno. Vedoucí práce Radek Štohl.
- [34] System 800xA - Control AC 800M - Getting Started. In: *ABB* [online]. Curych: ABB, 2011 [cit. 2019-10-30]. Dostupné z: https://library.e.abb.com/public/5d62dab1fb0cb932c125796d00208386/3BSE041880-510_A_en_System_800xA_Control_5.1_AC_800M_Getting_Started.pdf
- [35] System 800xA Engineering - Process Graphics 6.0. In: *ABB* [online]. Zurich: ABB, 2016 [cit. 2020-03-07]. Dostupné z: <https://search-ext.abb.com/library/Download.aspx?DocumentID=3BSE049230-600&LanguageCode=en&DocumentPartId=&Action=Launch>

- [36] Panel 800 version 6 - PP874. In: *ABB* [online]. Zurich: ABB, 2016 [cit. 2020-03-07]. Dostupné z: <https://search-ext.abb.com/library/Download.aspx?DocumentID=3BSE069458-600&LanguageCode=en&DocumentPartId=&Action=Launch>
- [37] Panel 800 Version 6, Panel Builder, Programming and Installation. In: *ABB* [online]. Zurich: ABB, 2016 [cit. 2020-03-07]. Dostupné z: <https://search-ext.abb.com/library/Download.aspx?DocumentID=3BSE069489-610&LanguageCode=en&DocumentPartId=&Action=Launch>
- [38] *ControlWeb* [online]. Zlín: Moravské přístroje a.s., 2019 [cit. 2019-07-21]. Dostupné z: <https://www.mii.cz/>

17 Seznam Příloh

Příloha A – Seznam použitých proměnných

Příloha B – Kompletní vývojový diagram

Příloha C – Pravdivostní tabulka automatického režimu

Příloha D – Uživatelská příručka

Příloha E – Infografika pro elektrický rozvaděč

Příloha F – Architektura DCS ABB 800xA